



**TUGAS AKHIR - TE 145561**

**PEMBUATAN PERANGKAT *MULTI-DISPLAY* YANG  
DIKONTROL SECARA TERPUSAT MENGGUNAKAN  
*WIRELESS***

Awataraning Tunjung Pradipta  
NRP 2214030049

Dosen Pembimbing  
Eko Pramunanto, ST., MT.

PROGRAM STUDI KOMPUTER KONTROL  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017





**FINAL PROJECT - TE 145561**

**MAKING *MULTI-DISPLAY* DEVICE WHICH IS  
CONTROLLED CENTRALLY USING *WIRELESS***

Awataraning Tunjung Pradipta  
NRP 2214030049

Advisor  
Eko Pramunanto, ST., MT.

COMPUTER CONTROL STUDY PROGRAM  
Electrical and Automation Engineering Department  
Vocational Faculty  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017



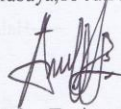
## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Pembuatan Perangkat *Multi-display* yang Dikontrol Secara Terpusat Menggunakan *Wireless***" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 18 Juli 2017



Awataraning Tunjung Pradipta  
NRP 2214030049

-----Halaman ini sengaja dikosongkan-----

**PEMBUATAN PERANGKAT *MULTI-DISPLAY* YANG  
DIKONTROL SECARA TERPUSAT MENGGUNAKAN  
*WIRELESS***

**TUGAS AKHIR**

Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Ahli Madya  
Pada  
Program Studi Komputer Kontrol  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing



Eko Pramananto, ST., MT.  
NIP. 19661203 199412 1 001

**SURABAYA  
JULI, 2017**

-----Halaman ini sengaja dikosongkan-----



# PEMBUATAN PERANGKAT *MULTI-DISPLAY* YANG DIKONTROL SECARA TERPUSAT MENGGUNAKAN *WIRELESS*

**Nama** : Awataraning Tunjung Pradipta  
**Pembimbing** : Eko Pramunanto, ST., MT.

## ABSTRAK

Dewasa ini, *Multi-display* sendiri digunakan untuk menampilkan suatu informasi. *Multi-display* disini disusun dari banyak layar televisi atau monitor untuk membentuk satu informasi yang utuh. Namun masih sedikit yang digunakan dalam ruang lingkup yang lebih kecil, yakni di sekolah atau di area perkantoran, karena di area tersebut juga membutuhkan suatu peyampaian informasi yang lebih efektif dan efisien.

Oleh karena itu, dibuatlah suatu tampilan *Multi-display dot matrix* P10 yang dapat dikontrol secara terpusat sehingga dapat menampilkan informasi. Pengontrolan secara terpusat yang dilakukan disini adalah menggunakan suatu komputer ataupun laptop yang sudah terhubung dengan modul *Wireless transmitter* yang terdiri atas Arduino dan XBee yang dapat digunakan untuk mengirimkan informasi menuju tampilan *Multi-display*. Sedangkan *Multi-display* disini sendiri, terhubung dengan modul *Wireless receiver* yang terdiri atas Arduino dan XBee yang digunakan untuk menerima informasi yang dikirimkan oleh kontrol pusat. Dimensi dari *Multi-display* ini adalah  $64 \times 32 \text{ cm}^2$  dengan besar bingkai  $72 \times 42 \times 8 \text{ cm}^3$ . Berdasar hasil pengujian, data yang dikirimkan dari PC/Laptop dan data yang diterima dan ditampilkan oleh *Multi-display* adalah sama. Jika data yang dikirim melebihi 63 karakter, kelebihannya tidak akan ditampilkan perangkat *Multi-display*. Kisaran waktu untuk menampilkannya berkisar antara 0,5 – 1,8 detik. Setelah menampilkan data yang dikirimkan, tampilannya akan kembali kepada tampilan *default*-nya, yaitu menampilkan jam dan tanggal, nama departemen serta nama ruang.

**Kata Kunci** : Arduino, *Multi-display* P10, XBee

-----Halaman ini sengaja dikosongkan-----

## **MAKING MULTI-DISPLAY DEVICE WHICH IS CONTROLLED CENTRALLY USING WIRELESS**

**Name : Awataraning Tunjung Pradipta**  
**Advisor : Eko Pramunanto, ST., MT.**

### **ABSTRACT**

*Today, Multi-display it self is used to display the information. Multi-display here composed of many television screen or monitor to form a complete information. But there is still little used in smaller scope, ie, at school or in the office area, because in these areas also need a peyampaian information more effectively and efficiently.*

*Therefore, they invented a Multi-display P10 dot matrix display that can be centrally controlled so as to display the information. Centrally controlling done here is to use a computer or laptop that is connected to the Wireless transmitter module consisting of Arduino and XBee can be used to transmit information to the multi-view display. While the Multi-display here alone, connected by Wireless receiver module which consists of an Arduino and XBee used for receiving information transmitted by the control center. The dimensions of this Multi-display is  $64 \times 32 \text{ cm}^2$  with a large frame  $72 \times 42 \times 8 \text{ cm}^3$ . Based on the test results, the data transmitted from the PC / Laptop and the data received and displayed by the Multi-display is the same. If the data that is sent exceeds 63 characters, the excess will not be displayed Multi-display devices. The range of time to display it ranged from 0.5 to 1.8 seconds. After displaying the data transmitted, the display will return to its default view, which displays the time and date, the department name and the name of the room.*

**Keywords : Arduino, Multi-display P10, XBee**

-----Halaman ini sengaja dikosongkan-----

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu diimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma 3 pada Program Studi Komputer Kontrol, Departemen Teknik Elektro Otomasi, Fakultas Teknik, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

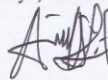
### **PEMBUATAN PERANGKAT *MULTI-DISPLAY* YANG DIKONTROL SECARA TERPUSAT MENGGUNAKAN *WIRELESS***

yang berguna sebagai sarana penyampaian pengumuman pada instansi dalam ruang lingkup yang lebih kecil seperti diarea perkantoran atau sekolah.

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Eko Pramunanto, ST., MT. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini, Moch Nafi Rizky Kurniawan atas kerjasama, semangat, dan dukungannya kepada penulis dalam penyelesaian Tugas Akhir ini. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 18 Juli 2017



Penulis

-----Halaman ini sengaja dikosongkan-----

# DAFTAR ISI

	HALAMAN
HALAMAN JUDUL .....	i
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
LEMBAR PENGESAHAN .....	vii
ABSTRAK.....	ix
<i>ABSTRACT</i> .....	xi
KATA PENGANTAR .....	xiii
DAFTAR ISI .....	xv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Permasalahan .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan.....	2
1.5 Metodologi Penelitian.....	2
1.6 Sistematika Laporan .....	5
1.7 Relevansi .....	5
BAB II TEORI DASAR .....	7
2.1 Modul <i>Dot Matrix Display</i> P10 .....	7
2.1.1 IC APM4953 .....	8
2.2 Mikrokontroler Arduino UNO .....	9
2.3 IDE .....	10
2.4 XBee S2C .....	11
2.5 <i>Power Supply Switching</i> .....	12
2.6 Modul <i>Real-Time Clock</i> (RTC) DS1307 .....	13
BAB III PERANCANGAN DAN PEMBUATAN ALAT .....	17
3.1 Blok Fungsional Sistem .....	17
3.2 Perancangan Perangkat Keras ( <i>Hardware</i> ).....	18
3.2.1 Perancangan <i>Multi-display</i> (Modul DMD P10) .....	18
3.2.2 Arduino UNO.....	20
3.2.3 XBee S2C.....	23
3.2.4 Konektor Modul <i>Real-Time Clock</i> DS1307 ke Arduino .	25
.....	25

3.2.5 Kotak atau Bingkai <i>Multi-display</i> .....	26
3.3 Perancangan Perangkat Lunak ( <i>Software</i> ) .....	27
3.3.1 Perancangan Program pada <i>Slave Multi-display</i> .....	29
BAB IV HASIL IMPLEMENTASI ALAT .....	33
4.1 Pengujian <i>Multi-display</i> .....	34
4.1.1 Pengujian Tegangan pada <i>Multi-display</i> .....	34
4.1.2 Pengujian Pengiriman Data .....	37
4.2 Pengujian Keseluruhan Sistem .....	46
4.2.1 Pengujian XBee Pada Program XCTU .....	47
4.2.2 Banyaknya Karakter yang Dapat Dikirimkan dan Kisaran Waktu dalam Menampilkannya .....	48
4.2.3 Jarak yang Dapat Dijangkau oleh XBee S2C .....	49
BAB V PENUTUP .....	51
5.1 Kesimpulan .....	51
5.2 Saran .....	51
DAFTAR PUSTAKA .....	53
LAMPIRAN .....	55
A.1. Dokumentasi Proses Masa Percobaan pada <i>Multi-display</i> ....	55
A.2. Dokumentasi Proses Perakitan <i>Multi-display</i> .....	56
A.3. Hasil <i>Monitoring</i> Karakter Melalui <i>Serial Monitor</i> .....	60
A.4. Perhitungan Waktu Tunda Pengiriman Karakter .....	63
A.5. Proses Pengiriman Data dari PC/Laptop ke <i>Multi-display</i> ...	63
A.6. Listing Program .....	65
A.7. Datasheets .....	71
RIWAYAT HIDUP PENULIS .....	83



## DAFTAR GAMBAR

	HALAMAN
Gambar 2. 1 DMD P10 Tampak Depan .....	7
Gambar 2. 2 DMD P10 Tampak Belakang .....	7
Gambar 2. 3 Modul DMD P10 (tampak belakang) .....	8
Gambar 2. 4 <i>Pinning</i> IC APM4953.....	8
Gambar 2. 5 Diagram Fungsional IC APM4953.....	9
Gambar 2. 6 Arduino UNO .....	10
Gambar 2. 7 Tampilan <i>Software</i> IDE.....	11
Gambar 2. 8 XBee S2C.....	12
Gambar 2. 9 <i>Power Supply</i> .....	12
Gambar 2. 10 Rangkaian Dalam <i>Power Supply Switching</i> .....	13
Gambar 2. 11 Bagian Dalam dari <i>Power Supply Switching</i> .....	13
Gambar 2. 12 Skema RTC DS1307 .....	14
Gambar 2. 13 Modul RTC DS1307 .....	15
Gambar 3. 1 Blok Fungsional Sistem.....	17
Gambar 3. 2 Rangkaian Dalam Modul DMD P10 .....	19
Gambar 3. 3 Sketsa <i>Multi-display</i> Tampak Depan.....	20
Gambar 3. 4 Sketsa <i>Multi-display</i> Tampak Belakang .....	20
Gambar 3. 5 Skematik dari Arduino UNO.....	21
Gambar 3. 6 <i>Pin-Out</i> Arduino UNO R3 .....	21
Gambar 3. 7 Konfigurasi Pin Arduino dan DMD P10 .....	23
Gambar 3. 8 <i>Port</i> Konektor Tampak Atas .....	23
Gambar 3. 9 <i>Port</i> Konektor Tampak Bawah.....	23
Gambar 3. 10 <i>Pin-Out</i> XBee.....	24
Gambar 3. 11 Integrasi XBee dengan Arduino .....	24
Gambar 3. 12 XBee <i>Shield</i> .....	25
Gambar 3. 13 Arduino + XBee + XBee <i>Shield</i> .....	25
Gambar 3. 14 Rangkaian RTC dengan Mikrokontroler .....	25
Gambar 3. 15 <i>Port</i> Konektor Tampak Atas .....	26
Gambar 3. 16 <i>Port</i> Konektor Tampak Bawah.....	26
Gambar 3. 17 Desain Bingkai untuk <i>Multi-display</i> .....	26
Gambar 3. 18 Hasil dari Rancangan Bingkai <i>Multi-display</i> .....	27
Gambar 3. 19 <i>Flowchart</i> Keseluruhan Sistem .....	29
Gambar 3. 20 <i>Flowchart</i> Program Pada <i>Slave</i> .....	31

Gambar 3. 21 Program Pada <i>Slave</i> .....	32
Gambar 4. 1 Hasil Perancangan <i>Multi-display</i> .....	33
Gambar 4. 2 Pengukuran Tegangan <i>Output</i> pada <i>Power Supply</i> .....	35
Gambar 4. 3 Pengukuran Tegangan <i>Input</i> pada Modul DMD P10 .....	36
Gambar 4. 4 Pengukuran Tegangan <i>Input</i> pada Kontroler Arduino ....	36
Gambar 4. 5 Hasil Ilustrasi Tampilan <i>Standby</i> pada <i>Multi-display</i> .....	38
Gambar 4. 6 Pengiriman Pengumuman Melalui <i>Serial Monitor</i> .....	39
Gambar 4. 7 Hasil Tampilan <i>Multi-display</i> .....	40
Gambar 4. 8 <i>Monitoring</i> pada <i>Slave</i> Melalui <i>Serial Monitor</i> .....	42
Gambar 4. 9 Penggunaan Aplikasi <i>Character Counter V6</i> .....	43
Gambar 4. 10 Menuliskan Data Sebanyak 70 Karakter.....	43
Gambar 4. 11 Hasil Tampilan <i>Multi-display</i> .....	44
Gambar 4. 12 Hasil <i>Monitoring</i> pada <i>Serial Monitor</i> .....	44
Gambar 4. 13 Pengujian dengan Program XCTU.....	47
Gambar 4. 14 Perangkat <i>Master</i> .....	48
Gambar 4. 15 Tampilan di Dalam Ruang, (a) <i>Slave</i> 1; (b) <i>Slave</i> 2.....	49

## DAFTAR TABEL

	HALAMAN
Tabel 2. 1 Spesifikasi Arduino UNO .....	10
Tabel 2. 2 Fungsi Pin RTC DS1307.....	14
Tabel 4. 1 Data Pengukuran Tegangan pada <i>Multi-display</i> .....	37
Tabel 4. 2 Hasil Pengujian Pengiriman Data.....	45
Tabel 4. 3 Pengujian Pengiriman Data Dengan Program XCTU.....	47
Tabel 4. 4 Hasil Pengujian Pengiriman Data Tanpa Halangan.....	49
Tabel 4. 5 Hasil Pengujian Pengiriman Data Dengan Halangan.....	50

-----Halaman ini sengaja dikosongkan-----

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Dewasa ini, teknologi *Multi-display* yang terdiri dari banyak layar atau monitor untuk membentuk sesuatu (baik itu gambar, video, informasi, dan sebagainya) sedang banyak digunakan. *Multi-display* ini banyak ditemukan pada jalan-jalan di kota besar, misalnya *videotron*. Teknologi panel *videotron* atau *Multi-display* ini sekarang sedang marak dikembangkan oleh beberapa perusahaan-perusahaan telekomunikasi besar, misalnya perusahaan Samsung.

Nyomananda, Presiden Direktur PT Media Indra Buana, memberi keterangan terkait pemasangan Samsung LED *Outdoor Panel Screen* di Jakarta. Samsung LED *Outdoor Panel Screen* sebagai yang terbesar di Indonesia dengan ukuran 450m<sup>2</sup> ini telah melalui uji kualitas terbaik sejak sebelum diberangkatkan, sehingga dapat menyajikan tampilan video yang lebih hidup. Selain Samsung LED XA100F, Samsung juga memiliki jajaran produk lainnya untuk layar panel LED, mengakomodir berbagai kebutuhan penggunaannya. Samsung memiliki tiga jenis Samsung "*Smart LED Signage*" yang diperuntukan untuk ruang dalam (*indoor*) dan untuk luar ruang (*outdoor*), yaitu "*Samsung in House*" (*indoor*), "*PrismView*" (*indoor* dan *outdoor*), dan "*JDM*" (*outdoor*). Layar panel untuk dalam ruangan tersedia dengan ukuran *pixel pitch* 1,5 sampai 16 *pixel pitch*, sedangkan untuk luar ruangan tersedia ukuran 6 sampai 25 *pixel pitch*. (netralnews.com,red)

Selain di jalan-jalan, *Multi-display* juga banyak ditemukan pada tempat pelayanan umum, sarana *game*/permainan, produksi audio profesional, dan lain-lain.

Fungsi utama *Multi-display* memang secara umum banyak digunakan untuk menampilkan suatu informasi. *Multi-display* yang digunakan adalah dari susunan LED, yang selanjutnya disebut *led matrix* atau *dot matrix*, yang diatur sedemikian rupa sehingga membentuk suatu tampilan dengan dimensi sesuai yang diinginkan. Umumnya, *dot matrix* sudah tersedia dalam bentuk modul sehingga tidak perlu bersusah payah untuk membuatnya lagi (selanjutnya modul ini disebut modul *dot matrix display*/DMD).

Dalam penggunaan *Multi-display* yang terdiri dari susunan modul *dot matrix*, banyak sekali pada ruang lingkup yang besar, seperti pada jalan raya. Namun sangat minim yang digunakan dalam ruang lingkup yang lebih kecil, seperti di tempat pendidikan (sekolah atau kampus) dan perkantoran. Oleh karena di ruang tersebut juga membutuhkan suatu perangkat sebagai penyalur informasi, maka disusunlah suatu pembuatan perangkat *Multi-display* yang dapat digunakan pada wilayah ruang lingkup yang kecil tersebut sebagai sarana untuk penyampaian informasi kepada setiap individu yang ada didalamnya. Nantinya, perangkat ini juga akan diintegrasikan dengan modul komunikasi *Wireless* agar dapat mengurangi pemakaian kabel, dan juga dengan tujuan dapat dikontrol secara terpusat agar informasinya berasal dari satu sumber yang dipercaya.

## **1.2 Permasalahan**

Permasalahan dalam Tugas Akhir ini adalah pembuatan perangkat *Multi-display* yang penggunaannya sudah banyak pada ruang lingkup yang besar, namun sangat minim sekali digunakan pada ruang lingkup yang lebih kecil, dalam hal ini di kampus. Alat ini akan dibuat sesuai dengan tempat dimana alat ini diletakkan, dan dapat menyampaikan pengumuman yang dibutuhkan secara jelas dan tepat.

## **1.3 Batasan Masalah**

Dalam pembuatan alat pada Tugas Akhir ini batasan masalah adalah pada *display* yang dikontrol secara terpusat yang hanya dapat menyampaikan informasi secara visual berbentuk teks.

## **1.4 Tujuan**

Tujuan penulisan Tugas Akhir ini adalah dapat membuat perangkat *Multi-display* dalam wilayah kampus yang sesuai dengan dimana alat ini diletakkan, dan dapat menampilkan informasi yang dibutuhkan melalui *Multi-display* secara jelas dan tepat.

## **1.5 Metodologi Penelitian**

Dalam pelaksanaan Tugas Akhir yang berupa Pembuatan Perangkat *Multi-display* yang Dikontrol Secara Terpusat Menggunakan *Wireless*, ada beberapa kegiatan yang dapat diuraikan sebagai berikut:

## **1. Study Pustaka dan Survei**

Pengumpulan data-data dari berbagai literatur mendukung yang dalam proses pembuatan Tugas Akhir ini. Setelah itu melakukan survei di mana alat ini dapat di pasang pada Departemen Teknik Elektro Otomasi.

## **2. Tahap Persiapan**

Pada tahap ini akan dilakukan studi literatur mengenai:

- a. Mempelajari macam-macam *Multi-display*.
- b. Mempelajari sistem kerja dari modul *Multi-display* yang dipilih, modul DMD P10.
- c. Mempelajari pemrograman IDE Arduino.

## **3. Tahap Identifikasi dan Pemodelan Alat**

Pada tahap ini akan dilakukan identifikasi dari alat sesuai data yang telah didapatkan dari studi literatur serta dilakukan pemodelan dari alat yang akan dikerjakan.

## **4. Tahap Perancangan Alat**

### **a. Object Penelitian**

Sesuai dengan tujuan Tugas Akhir yang telah diuraikan maka objek penelitian dalam penyusunan Tugas Akhir ini meliputi :

- a. Desain dari *Multi-display* dengan menggunakan modul P10.
- b. Tampilan yang dihasilkan dari *Multi-display*.

### **b. Tempat Perancangan**

Tempat kegiatan Tugas Akhir ini dilakukan penelitian dan perancangan di:

- a. Lab BB 102 Laboratorium Elektronika Dasar
- b. Rumah Pribadi di Jalan Medokan Semampir Indah No. 117
- c. Rumah Kos di Jalan Keputih Gang 2B No. 25

### **c. Waktu Perancangan**

Kegiatan Tugas Akhir ini dilakukan penelitian dan perancangan Mulai dari Bulan Februari 2017 sampai dengan Bulan Juni 2017.

### **d. Alat Terkait**

Beberapa alat yang digunakan dalam Tugas Akhir ini diantaranya sebagai berikut:

- a. XBee S2C
- b. Arduino UNO
- c. Modul DMD P10

- d. Modul *Real-Time Clock* DS1307
- e. *Power Supply Switching* 5V 10A

## 5. Tahap Pembuatan Alat

Perancangan *hardware* yang dilakukan dengan merancang rangkaian-rangkaian penyusun *Multi-display* yang akan di gunakan dalam masing-masing *Slave* meliputi:

- a. Rancangan tata letak dari *Multi-display*.
- b. Konfigurasi Arduino+ XBee + RTC + P10, koneksi per modul P10, dan koneksi untuk *power supply*.

Serta perancangan *software* meliputi:

- a. Menyusun *flowchart* secara keseluruhan dari sistem yang akan dibuat.
- b. Menyusun *flowchart* untuk *slave* agar dapat dibuat suatu program yang sesuai dengan yang diinginkan.

## 6. Tahap Pengujian dan Analisa

Dalam proses uji coba dan analisis data terhadap alat diharapkan akan membantu dan mempermudah mahasiswa apabila alatnya mengalami kesalahan atau kegagalan. Tahapan ini dilakukan dengan cara melakukan pengecekan alat. Setelah dilakukan, melakukan analisis terhadap kesalahan dan kegagalan apabila terjadi, lalu dilakukan pemecahan masalah terhadap permasalahan tersebut. Beberapa pengujian yang akan dilakukan adalah sebagai berikut:

- a. Pengukuran tegangan *input* yang didapatkan oleh kontroler (Arduino) dan modul DMD P10, serta pengukuran tegangan *output* pada *power supply*.
- b. Ketepatan informasi yang ditampilkan oleh *Multi-display* P10.
- c. Banyaknya jumlah karakter yang dapat ditampilkan oleh *Multi-display*.
- d. Perhitungan penundaan waktu (*delay*) dalam pengiriman data.
- e. Monitoring yang dilakukan melalui *serial monitor* pada *software* IDE.

## 7. Tahap Penyusunan Laporan

Setelah alat berhasil dibuat dan bekerja dengan baik tanpa adanya *error*, pengambilan data dan analisa data terpenuhi, maka tahap selanjutnya yaitu penyusunan laporan untuk buku Tugas Akhir. Diharapkan buku Tugas Akhir ini bermanfaat bagi semua orang, dan



dapat dijadikan pedoman dalam melanjutkan dan mengembangkan ide Tugas Akhir ini.

## **1.6 Sistematika Laporan**

Untuk pembahasan lebih lanjut, laporan Tugas Akhir ini disusun dengan sistematika sebagai berikut:

### **Bab I PENDAHULUAN**

Membahas tentang latar belakang, permasalahan, batasan masalah, tujuan, metodologi, sistematika laporan, serta relevansi Tugas Akhir yang dibuat.

### **Bab II TEORI DASAR**

Menjelaskan teori yang berisi teori-teori dasar yang dijadikan landasan dan mendukung dalam perencanaan dan pembuatan alat yang dibuat.

### **Bab III PERANCANGAN DAN PEMBUATAN ALAT**

Membahas perencanaan dan pembuatan tentang perencanaan dan pembuatan *hardware* yang meliputi DMD P10, Arduino UNO, XBee S2C, Modul RTC, *Power Supply* dan Bingkai *Multi-display*, dan perancangan *software* yang meliputi program yang akan digunakan untuk menjalankan alat tersebut.

### **Bab IV HASIL IMPLEMENTASI ALAT**

Membahas pengujian alat dan menganalisa data yang didapat dari pengujian tersebut serta membahas tentang pengukuran, pengujian, dan penganalisaan terhadap alat.

### **Bab V PENUTUP**

Berisi penutup yang menjelaskan tentang kesimpulan yang didapat dari Tugas Akhir ini dan saran-saran untuk pengembangan alat ini lebih lanjut.

## **1.7 Relevansi**

Dari pembuatan alat ini diharapkan akan tercipta beberapa manfaat yaitu:

- a. Mempermudah pengontrolan *Multi-display* dari jarak jauh.

- b. Mempermudah penyaluran informasi.
- c. Dapat digunakan dalam radius yang cukup jauh (tergantung dari kemampuan suatu komponen *Wireless* yang digunakan).
- d. Memperoleh data informasi yang akurat dan terpercaya.

## BAB II

### TEORI DASAR

Beberapa teori penunjang yang dipaparkan dalam buku Tugas Akhir ini adalah teori dasar mengenai proses penulisan apa-apa saja komponen atau alat yang digunakan dalam membuat dan merancang Tugas Akhir ini.

#### 2.1 Modul *Dot Matrix Display* P10

*Dot matrix display* (sering disingkat menjadi istilah DMD) adalah sebuah LED *display* yang dirancang dengan ukuran 16x64 yang dapat digunakan dalam luar ruang maupun dalam ruang. Pada *display* ini dapat disambungkan dengan *display* P10 lainnya dengan paralel maupun seri. Pada *display* ini dapat digunakan masukkan dari *power supply* maupun dari Arduino langsung yang terhubung dengan PC. Penggunaan *display* ini banyak digunakan untuk *running text*. Pada **Gambar 2. 1** dan **Gambar 2. 2** diberikan tampilan DMD P10 Tampak Depan dan Tampak Belakang.



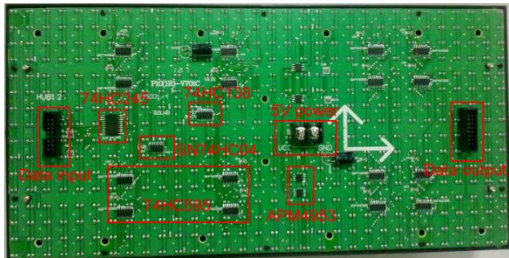
**Gambar 2. 1** DMD P10 Tampak Depan



**Gambar 2. 2** DMD P10 Tampak Belakang

*Running text* adalah papan iklan promosi dan informasi yang terbuat dari LED yang dapat diprogram menggunakan komputer untuk menampilkan tulisan, gambar dan animasi berwarna. Pada umumnya, *running text* dibedakan menjadi dua, yaitu *running text outdoor* dan *indoor*. *Running text outdoor* adalah *running text* yang diletakkan diluar ruangan dan biasanya menggunakan jenis LED DIP (bentuk *led* yang benjol-benjol), sedangkan *running text indoor* adalah *running text* yang diletakkan di dalam ruangan dan biasanya menggunakan jenis LED *dot matrix* (bentuk *led* yang permukaannya datar).

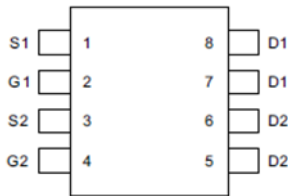
Dalam satu modul P10 ini, terdapat beberapa IC yang mempunyai fungsi masing-masing. Kelima IC tersebut adalah IC 74HC245, IC 74HC138, IC SN74HC04, IC 74HC595, dan IC APM4953. Dibawah diberikan letak kelima IC tersebut pada **Gambar 2. 3** pada modul DMD P10.



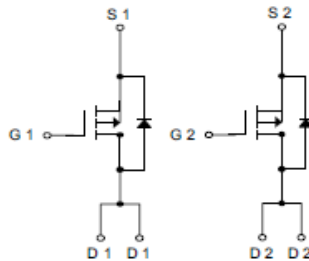
**Gambar 2. 3** Modul DMD P10 (tampak belakang)

### 2.1.1 IC APM4953

APM4953 adalah salah satu jenis dari *mode enhancement* MOSFET tipe *dual P-channel*. E-MOSFET (*Enhancement-metal-oxide semiconductor* FET) adalah MOSFET tipe peningkatan yang terdiri dari E-MOSFET kanal-P dan E-MOSFET kanal-N. E-MOSFET kanal-P dan E-MOSFET kanal-N pada dasarnya sama, yang berbeda hanyalah polaritas pada pemberian biasnya saja. Aplikasi dari IC ini adalah pengelolaan daya pada komputer *notebook*, peralatan portabel, dan sistem bertenaga baterai. Untuk *Pinning* dan Diagram Fungsional dari IC APM4953 diberikan pada **Gambar 2. 4** dan **Gambar 2. 5**.



**Gambar 2. 4** *Pinning* IC APM4953



**Gambar 2. 5** Diagram Fungsional IC APM4953

## 2.2 Mikrokontroler Arduino UNO

Mikrokontroler adalah sebuah sistem komputer fungsional dalam sebuah *chip*. Di dalamnya terkandung sebuah inti prosesor, memori (sejumlah kecil RAM, memori program, atau keduanya), dan perlengkapan input output. Dengan kata lain, mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja mikrokontroler sebenarnya membaca dan menulis data. Mikrokontroler merupakan komputer didalam chip yang digunakan untuk mengontrol peralatan elektronik, yang menekankan efisiensi dan efektifitas biaya. Secara harfiahnya bisa disebut “pengendali kecil” dimana sebuah sistem elektronik yang sebelumnya banyak memerlukan komponen-komponen pendukung seperti IC TTL dan CMOS dapat direduksi/diperkecil dan akhirnya terpusat serta dikendalikan oleh mikrokontroler ini.

Arduino merupakan rangkaian elektronik yang bersifat *open source*, serta memiliki perangkat keras dan lunak yang mudah untuk digunakan. Arduino dapat mengenali lingkungan sekitarnya melalui berbagai jenis sensor dan dapat mengendalikan lampu, motor, dan berbagai jenis aktuator lainnya. Arduino mempunyai banyak jenis, di antaranya Arduino Uno, Arduino Mega 2560, Arduino Fio, dan lainnya.

Arduino UNO adalah sebuah board mikrokontroler yang didasarkan pada ATmega328. Arduino UNO mempunyai 14 pin digital *input/output* (6 di antaranya dapat digunakan sebagai *output* PWM), 6 *input* analog, sebuah osilator kristal 16 MHz, sebuah koneksi USB, sebuah *power jack*, sebuah ICSP *header*, dan sebuah tombol *reset*.

**Gambar 2. 6** menunjukkan gambar dari Arduino UNO.



**Gambar 2. 6** Arduino UNO

Untuk meringkasnya, diberikan **Tabel 2. 1** dibawah sebagai ringkasan dari penjelasan Arduino UNO.

**Tabel 2. 1** Spesifikasi Arduino UNO

Mikrokontroler	ATmega328
Tegangan pengoperasian	5V
Tegangan <i>input</i> yang disarankan	7-12V
Batas tegangan <i>input</i>	6-20V
Jumlah pin I/O digital	14 (6 di antaranya menyediakan keluaran PWM)
Jumlah pin <i>input</i> analog	6
Arus DC tiap pin I/O	40 mA
Arus DC untuk pin 3.3V	50 mA
Memori <i>Flash</i>	32 KB (ATmega328), sekitar 0.5 KB digunakan oleh <i>bootloader</i>
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
<i>Clock Speed</i>	16 MHz

“Uno” berarti satu dalam bahasa Italia dan dinamai untuk menandakan keluaran (produk) Arduino 1.0 selanjutnya. Arduino UNO dan versi 1.0 akan menjadi referensi untuk versi-versi Arduino selanjutnya. Arduino UNO adalah sebuah seri terakhir dari *board* Arduino USB dan model referensi untuk papan Arduino, untuk suatu perbandingan dengan versi sebelumnya.

### 2.3 IDE

IDE itu merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Tampilan dari *software* IDE Arduino ini diberikan pada **Gambar 2. 7**. Arduino menggunakan bahasa

pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama *bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler. Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan *library* C/C++ yang biasa disebut *wiring* yang membuat operasi *input* dan *output* menjadi lebih mudah. Arduino IDE ini dikembangkan dari *software processing* yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino.



**Gambar 2. 7** Tampilan *Software IDE*

## 2.4 XBee S2C

XBee adalah teknologi yang dikembangkan untuk memantau dan merekam parameter. Sistem ini terdiri dari PIC mikrokontroler untuk penanganan sensor dan sinyal dikomunikasikan melalui XBee *transmitter* dan XBee *receiver*. XBee berbasis nirkabel, perangkat ini digunakan sebagai komunikasi nirkabel jarak antara modul. Informasi yang ditransmisikan titik dengan titik menggunakan XBee *transmitter* dan *receiver* dan dikirim ke *server* modul untuk memeriksa keadaan parameter. Perangkat XBee S2C ditunjukkan pada **Gambar 2. 8**.



**Gambar 2. 8** XBee S2C

## 2.5 Power Supply Switching

*Power Supply* atau dalam bahasa Indonesia disebut dengan Catu Daya adalah suatu alat listrik yang dapat menyediakan energi listrik untuk perangkat listrik ataupun elektronika lainnya. Pada dasarnya *Power Supply* atau Catu Daya ini memerlukan sumber energi listrik yang kemudian mengubahnya menjadi energi listrik yang dibutuhkan oleh perangkat elektronika lainnya. Oleh karena itu, *Power Supply* kadang-kadang disebut juga dengan istilah *Electric Power Converter*.

*Power Supply* dapat dibedakan menjadi beberapa jenis, diantaranya adalah DC *Power Supply*, AC *Power Supply*, *Switch-Mode Power Supply*, *Programmable Power Supply*, *Uninterruptible Power Supply*, *High Voltage Power Supply*. Pada **Gambar 2. 9**, diberikan contoh gambar dari *power supply*.



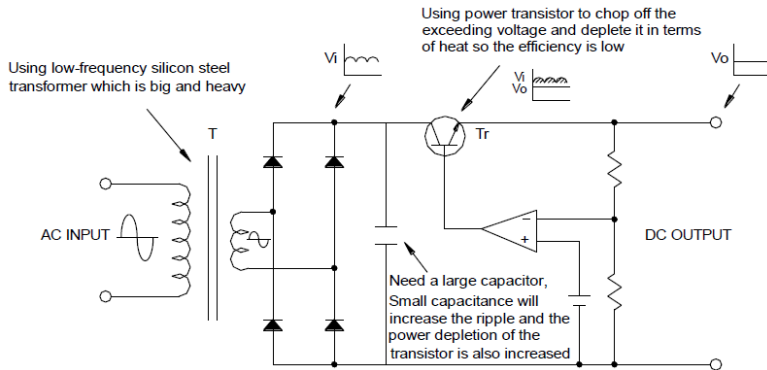
**Gambar 2. 9** Power Supply

Yang digunakan pada tugas ini adalah *Switch-Mode Power Supply* atau disebut juga *Power Supply Switching*. Dimana, *Switch-Mode Power Supply* adalah jenis *Power Supply* yang langsung menyearahkan (*rectify*) dan menyaring (*filter*) tegangan *input* AC untuk mendapatkan tegangan DC. Tegangan DC tersebut kemudian di-*switch* ON dan OFF pada frekuensi tinggi dengan sirkuit frekuensi tinggi sehingga menghasilkan arus AC yang dapat melewati Transformator Frekuensi Tinggi.

Pada bagian ini, digunakan *Power Supply Switching* dengan tegangan keluaran 5V dan arus sebesar 10A dimana merupakan sumber untuk menyalakan *Multi-display* dan juga untuk catu daya kontroler Arduino. Catu daya dengan spesifikasi ini digunakan karena untuk



menyalakan *Multi-display* dan Arduino cukup menggunakan sumber dengan tegangan keluaran 5V. Pada masing masing panel memiliki *power*-nya masing-masing. Untuk *power supply* yang digunakan dapat menyalakan setidaknya 4 modul DMD P10 dengan cahaya pancaran cukup terang. Rangkaian dalam dan bagian dalam dari *Power Supply Switching* ini diberikan berturut-turut pada **Gambar 2. 10** dan **Gambar 2. 11**.



**Gambar 2. 10** Rangkaian Dalam *Power Supply Switching*



**Gambar 2. 11** Bagian Dalam dari *Power Supply Switching*

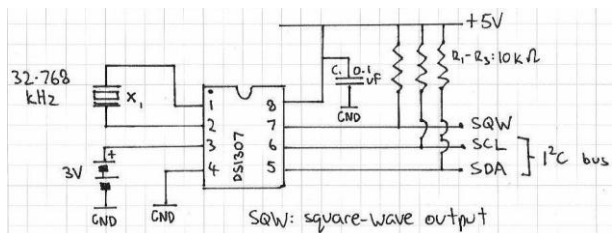
## 2.6 Modul *Real-Time Clock* (RTC) DS1307

*Real-time clock* disingkat RTC adalah jam di komputer yang umumnya berupa sirkuit terpadu yang berfungsi sebagai pemelihara waktu. RTC umumnya memiliki catu daya komputer (umumnya berupa

baterai litium) sehingga dapat tetap berfungsi ketika catu daya komputer terputus. Kebanyakan RTC menggunakan *oscillator* kristal.

Komponen *real-time clock* adalah komponen IC penghitung yang dapat difungsikan sebagai sumber data waktu baik berupa data jam, hari, bulan maupun tahun. Komponen DS1307 berupa IC yang perlu dilengkapi dengan komponen pendukung lainnya seperti *crystal* sebagai sumber *clock* dan *Battery External* 3 volt sebagai sumber energi cadangan agar fungsi penghitung tidak berhenti.

Sistem RTC DS1307 memerlukan baterai eksternal 3V yang terhubung ke pin Vbat dan *ground*. Pin X1 dan X2 dihubungkan dengan kristal osilator 32,768 KHz. Sedangkan pin SCL, SDA, dan SQW/OUT di-*pull-up* dengan resistor (nilainya 1k s.d 10k) ke Vcc. Skema dari RTC DS1307 diberikan pada **Gambar 2. 12** beserta penjelasan pin-nya pada **Tabel 2. 2**.



**Gambar 2. 12** Skema RTC DS1307

**Tabel 2. 2** Fungsi Pin RTC DS1307

No.	Nama Pin	Fungsi
1	X1	Oscillator Crystal
2	X2	32.768 KHz
3	VBAT	Battery Input (+3V)
4	GND	Ground
5	SDA	Serial Data
6	SCL	Serial Clock Input
7	SQW/OUT	Square Wave/Output Driver
8	VCC	Supply Power

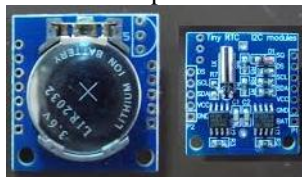
Keterangan:

- Pin Vcc (Nomor 8) berfungsi sebagai sumber energi listrik utama. Tegangan kerja dari komponen ini adalah 5V, dan ini sesuai dengan tegangan kerja dari mikrokontroler papan Arduino.

- Pin GND (Nomor 4) harus dihubungkan *ground* yang dimiliki oleh komponen RTC dengan *ground* dari *battery back-up*.
- SCL berfungsi sebagai saluran *clock* untuk komunikasi data antara mikrokontroler dengan RTC.
- SDA berfungsi sebagai saluran data untuk komunikasi data antara mikrokontroler dengan RTC.
- X1 dan X2 berfungsi untuk saluran *clock* yang bersumber dari *crystal external*.
- Vbat berfungsi sebagai saluran energi listrik dari *battery external*.

RTC tipe DS1307 merupakan jenis pewaktu yang menggunakan komunikasi serial untuk operasi tulis baca, dengan gambar modul yang ditunjukkan pada **Gambar 2. 13** dan spesifikasi berikut ini:

- *Real-time clock* (RTC) menyimpan data-data detik, menit, jam, tanggal, bulan, hari dalam seminggu, dan tahun valid hingga 2100;
- 56-byte, *battery-backed*, RAM *nonvolatile* (NV) RAM untuk penyimpanan;
- Antarmuka serial *two-wire* (I2C);
- Sinyal luaran gelombang-kotak terprogram (*programmable squareware*);
- Deteksi otomatis kegagalan-daya (*power-fail*) dan rangkaian *switch*;
- Konsumsi daya kurang dari 500 nA menggunakan mode baterai cadangan dengan operasional *oscillator*.
- Tersedia fitur industri dengan ketahanan suhu: -40°C hingga +85°C;
- Tersedia dalam kemasan 8-pin DIP atau SOI.



**Gambar 2. 13** Modul RTC DS1307

-----Halaman ini sengaja dikosongkan-----

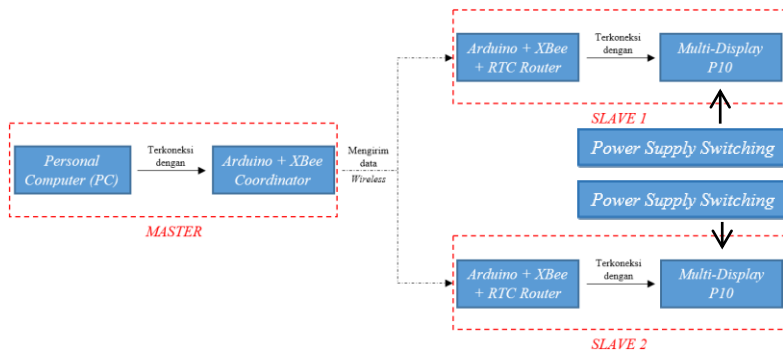
## BAB III

### PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini akan dijelaskan mengenai perancangan dan pembuatan Pembuatan Perangkat *Multi-display* yang Dikontrol Secara Terpusat Menggunakan *Wireless* meliputi blok fungsional sistem yang akan menjelaskan proses kerja alat dalam bentuk alur diagram, dan perancangan. Dalam perancangan ini terdapat 2 tahap perancangan, yaitu perancangan *hardware* dan *software*.

#### 3.1 Blok Fungsional Sistem

Sebelum melakukan pembuatan alat yang meliputi perangkat keras (*hardware*) dan perangkat lunak (*software*), diperlukan sebuah perencanaan alat berupa blok diagram yang dapat dilihat pada gambar dibawah ini.



**Gambar 3. 1** Blok Fungsional Sistem

Dari **Gambar 3.1**, dijelaskan bahwa dalam Tugas Akhir ini, akan dibuat sebuah *Multi-display* yang di mana dalam satu panel terdiri dari empat *dot matrix display* P10. Pada perancangan dari Tugas Akhir ini digunakan XBee sebagai media transmisi nirkabel (*Wireless*) dari PC (*Personal Computer*) ke tiap-tiap *Multi-display*. Untuk dapat melakukan komunikasi secara *Wireless*, XBee diatur sedemikian rupa sehingga dapat digunakan peralatan penerima dan pengirim. Jaringan *Wireless* memiliki keunggulan dan keuntungan dibanding dengan jaringan kabel. Mobilitas jaringan *Wireless* menyediakan pengaksesan kepada pengguna dimana saja, selama berada dalam batas spesifikasi dari alat *Wireless* tersebut. *Operator* PC ( pada blok

*Master*) bisa mengatur pengumuman yang ingin disampaikan pada *display* berupa *running text* secara *real-time* yang mana pengumuman tersebut bersifat sama pada tiap *display*-nya. Pengiriman ini menggunakan komunikasi antara XBee yang ada pada *master* dan XBee yang ada pada tiap *Slave*. Jika tidak ada pengumuman, ada fitur untuk keadaan *standby* maka *display* akan mengeluarkan *static text* yang akan menunjukkan tempat dan juga waktu *display* itu berada.

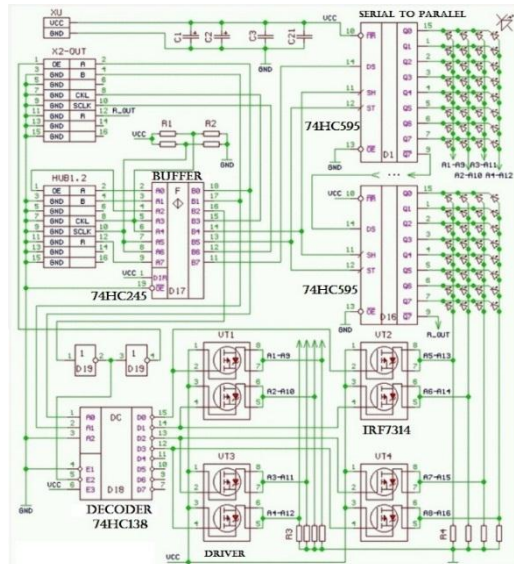
Sumber tegangan dari blok *Master*, kontroler yang terdiri dari Arduino dan XBee langsung mendapatkan sumber tegangan dari PC. Sedangkan, pada tiap *Multi-display* mendapatkan sumber tegangan dari *power supply switching* sebesar 5V.

### **3.2 Perancangan Perangkat Keras (*Hardware*)**

Dalam subbab ini, akan dibahas mengenai perancangan perangkat keras dari alat yang mendukung seluruh perancangan dan pembuatan alat yang meliputi *Multi-display* yang tersusun atas 4 LED modul P10 yang tersusun 2 baris dan 2 kolom, kontroler yang terdiri dari Arduino UNO R3 yang terhubung dengan XBee S2C dan modul RTC, dan *Power Supply Switching* 5V 10A serta kotak atau bingkai *Multi-display*.

#### **3.2.1 Perancangan *Multi-display* (Modul DMD P10)**

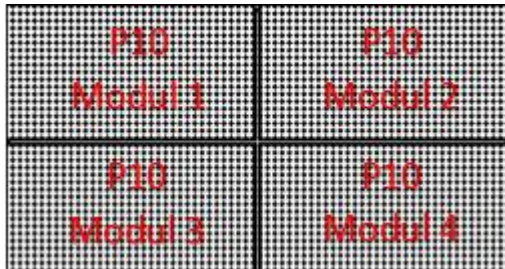
Perancangan *Multi-display* ini tersusun atas 4 modul DMD P10 dan bingkai atau kotak untuk menempatkan modul P10 tersebut. Seperti yang telah dijelaskan diatas, perancangan *Multi-display* disini menggunakan modul DMD yang diberi nama modul P10. Penamaan P10 pada *display* ini maksudnya adalah jarak dari *led* ke *led* yang lain ada 10 mm. Ini didasarkan pada arti dari Pxx sendiri adalah *Pixel Pitch*. Maka dari itu, ada banyak sekali tipe dari *display* ini diantaranya P10, P16, P8, P3,75, dan lain-lain. Modul ini dapat disusun menjadi 4 dimana terdiri dari 2 kolom dan 2 baris. Digunakannya modul DMD P10 pada Tugas Akhir ini dikarenakan modul ini sesuai dengan apa yang diharapkan oleh penulis, yaitu mempunyai dimensi 32x16, tidak terlalu besar dan tidak terlalu kecil, dan juga memudahkan dalam proses peng-koneksi-an atau penyambungannya dengan modul P10 yang lain. Ini dapat dijelaskan dari rangkaian dalam modul P10 ini pada **Gambar 3. 2**.



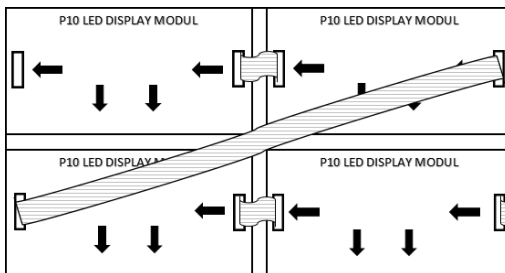
**Gambar 3. 2** Rangkaian Dalam Modul DMD P10

Penjelasannya, data akan dikirim dari arduino secara serial melalui soket HUB1.2 kemudian diterima oleh IC *serial-to-parallel* 74595. Jika ada tambahan modul akan diambil dari keluaran 74595 yang terakhir yang dihubungkan ke soket X2-OUT. Untuk *supply* arus diberikan oleh IC *driver* penguat daya VT1,VT2..dst. Untuk memilih kolom mana yang menyala diaktifkan oleh IC *demultiplexer*.

Dari gambar diatas, secara sederhana, jika ingin menambahkan modul P10 lagi, dapat ditambahkan melalui “*data output*” led modul pertama ke “*data input*” led modul kedua dan seterusnya menggunakan kabel *ribbon*. Sehingga, jika digambarkan sketsanya maka akan membentuk sketsa bagian depan yang ditunjukkan pada **Gambar 3. 3** dan bagian belakang pada **Gambar 3. 4**.



**Gambar 3. 3** Sketsa *Multi-display* Tampak Depan



**Gambar 3. 4** Sketsa *Multi-display* Tampak Belakang

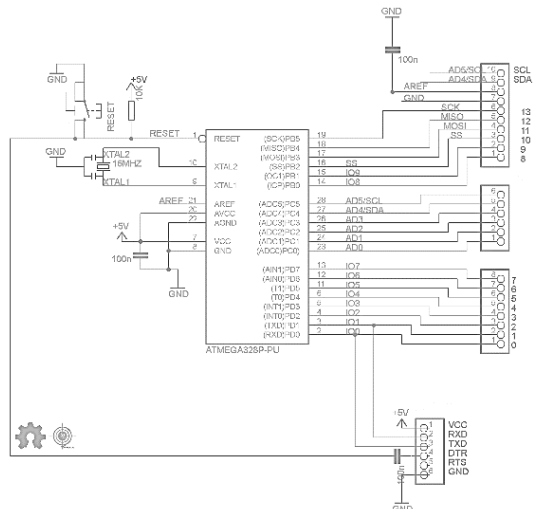
### 3.2.2 Arduino UNO

Untuk kontroler yang akan digunakan adalah kontroler Arduino UNO. Arduino UNO adalah sebuah *board* mikrokontroler yang didasarkan pada ATmega328. Arduino UNO mempunyai 14 pin digital *input/output* (6 di antaranya dapat digunakan sebagai *output* PWM), 6 *input* analog, sebuah osilator kristal 16 MHz, sebuah koneksi USB, sebuah *power jack*, sebuah ICSP *header*, dan sebuah tombol *reset*.

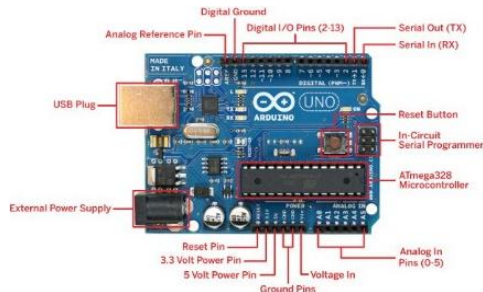
Arduino UNO adalah alat yang diperlukan untuk mendukung sebuah rangkaian mikrokontroler. Cukup dengan menghubungkannya ke komputer dengan kabel USB atau diberi *power* dengan adaptor AC-DC atau baterai, Arduino UNO sudah dapat digunakan tanpa khawatir akan melakukan sesuatu yang salah.

Kontroler ini digunakan karena spesifikasi yang dibutuhkan untuk melakukan Tugas Akhir ini sudah cukup, seperti pemakaian pin digital, I2C, dan lain-lain. Selain itu, kontroler ini telah umum dan banyak sumber atau referensi yang didapatkan jika menggunakan kontroler ini. Arduino UNO R3 memiliki konfigurasi skematik pada **Gambar 3. 5** serta *pin-out* pada **Gambar 3. 6**.





### Gambar 3. 5 Skematik dari Arduino UNO



**Gambar 3. 6** *Pin-Out* Arduino UNO R3

Seperti yang telah disebutkan sebelumnya, Arduino UNO memiliki 14 buah digital pin yang dapat digunakan sebagai *input* atau *output*, sengan menggunakan fungsi *pinMode()*, *digitalWrite()*, dan *digitalRead()*. Pin-pin tersebut bekerja pada tegangan 5V, dan setiap pin dapat menyediakan atau menerima arus 20mA, dan memiliki tahanan *pull-up* sekitar 20-50Kohm (secara *default* dalam posisi *disconnect*). Nilai maksimum adalah 40mA, yang sebisa mungkin dihindari untuk menghindari kerusakan *chip* mikrokontroler.

Beberapa pin memiliki fungsi khusus:

- *Serial*, terdiri dari 2 pin : pin 0 (RX) dan pin 1 (TX) yang digunakan untuk menerima (RX) dan mengirim (TX) data serial.
- *External Interrupt*, yaitu pin 2 dan pin 3. Kedua pin tersebut dapat digunakan untuk mengaktifkan *interrupt*. Gunakan fungsi *attachInterrupt()*.
- *PWM* : Pin 3, 5, 6, 9, 10, dan 11 menyediakan *output* PWM 8-bit dengan menggunakan fungsi *analogWrite()*.
- *SPI* : Pin 10 (SS), 11 (MOSI), 12 (MISO), dan 13 (SCK) mendukung komunikasi SPI dengan menggunakan *SPI Library*.
- *LED* : Pin 13. Pada pin 13 terhubung *built-in led* yang dikendalikan oleh digital pin nomor 13.
- *TWI* : Pin A4 (SDA) dan pin A5 (SCL) yang mendukung komunikasi TWI dengan menggunakan *Wire Library*.

Arduino UNO memiliki 6 buah input analog, yang diberi tanda dengan A0, A1, A2, A3, A4, A5. Masing-masing pin analog tersebut memiliki resolusi 10 bits (jadi bisa memiliki 1024 nilai). Secara *default*, pin-pin tersebut diukur dari *ground* ke 5V, namun bisa juga menggunakan pin AREF dengan menggunakan fungsi *analogReference()*. Beberapa pin lainnya pada board ini adalah :

- AREF, sebagai referensi tegangan untuk *input* analog.
- *Reset*; hubungkan ke LOW untuk melakukan *reset* terhadap mikrokontroler. Sama dengan penggunaan tombol *reset* yang tersedia.

Jika kontroler yang akan digunakan adalah Arduino UNO, untuk dapat terhubung dengan DMD P10 (atau dalam hal ini *Multi-display*), perlu dilakukan pembuatan *port* konektor atau penghubung antara Arduino dan *Multi-display*, yang selanjutnya disebut sebagai *DMDP10-to-Arduino Connector*. Pembuatan *port* yang digunakan untuk menghubungkan DMD P10 dan Arduino UNO mengikuti konfigurasi pada **Gambar 3. 7**.

OE-D9	1	2	A-D6
GND	3	4	B-D7
	5	6	
	7	8	CLK-D13
	9	10	SCLK-D8
	11	12	DATA-D11
	13	14	
	15	16	

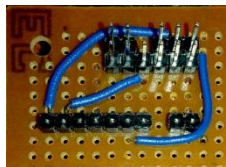
Kabel P10	Arduino
Pin 1	Pin Digital 9
Pin 2	Pin Digital 6
Pin 3	GND
Pin 4	Pin Digital 7
Pin 8	Pin Digital 13
Pin 10	Pin Digital 8
Pin 12	Pin Digital 11

**Gambar 3. 7** Konfigurasi Pin Arduino dan DMD P10

Penjelasan pin pada port konektor:

- OE: Output Enable untuk on/off semua LED
- A dan B: memilih kolom yg aktif.
- CLK: SPI clock
- SCLK: Latch data register
- Data: SERIAL DATA SPI

Hasil dari pembuatan dengan *pcb dot*, ditampilkan dengan hasil solderan yang ditunjukkan **Gambar 3. 8** untuk bagian atas dan **Gambar 3. 9** untuk bagian bawah.



**Gambar 3. 8** Port Konektor Tampak Atas

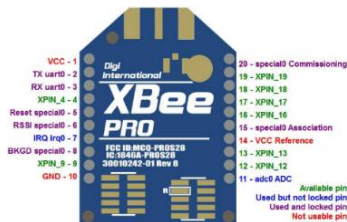


**Gambar 3. 9** Port Konektor Tampak Bawah

### 3.2.3 XBee S2C

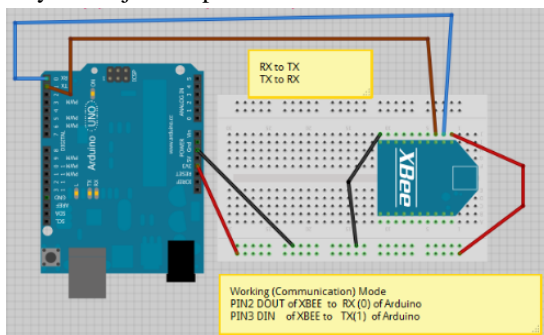
Dalam Tugas Akhir ini, kontroler Arduino UNO ini diintegrasikan dengan suatu modul nirkabel yang disebut sebagai XBee. XBee adalah modul *Wireless* atau nirkabel masa kini yang digunakan sebagai

transmisi jarak jauh untuk mengirimkan dan menerima data. Untuk referensi dalam penggunaan modul ini minim sekali, sehingga yang dapat dilakukan adalah hanya mengirimkan karakter yang berisi suatu program didalamnya (karena referensi yang didapatkan berkisar antara operasi ON/OFF suatu komponen) dan pengiriman *string* untuk ditampilkan. Untuk konfigurasi pin dari XBee diberikan pada **Gambar 3. 10.**



**Gambar 3. 10** Pin-Out XBee

Dalam mengintegrasikan XBee dengan Arduino, perlu diperhatikan letak dari pin Rx dan Tx baik pada XBee maupun Arduino. Pin Rx pada XBee (Pin 3; DIN) terhubung dengan Pin 0 (Tx) pada Arduino dan Pin Tx pada XBee (Pin 2; DOUT) terhubung dengan Pin 1 (Rx) pada Arduino. Sehingga jika digambarkan *wiring* atau pengkabelannya ditunjukkan pada **Gambar 3. 11.**



**Gambar 3. 11** Integrasi XBee dengan Arduino

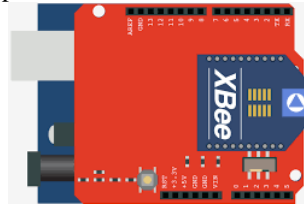
Untuk mempermudah dalam penyambungan antara XBee dengan Arduino, maka diciptakan suatu papan yang disebut dengan XBee *Shield*. XBee *Shield* adalah suatu papan yang mengizinkan Arduino untuk berkomunikasi secara *Wireless* menggunakan Zigbee/XBee. Bentuk dari XBee *Shield* ini menyerupai *pin header female* pada

Arduino untuk penggunaan pada pin 2 sampai 7 dan analog *input*, yang mana di tutupi oleh *shield*, yang ditunjukkan pada **Gambar 3. 12**.



**Gambar 3. 12** XBee Shield

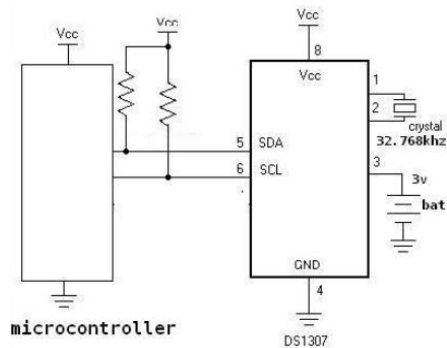
Sehingga, jika dipasangkan antara Arduino, XBee dan Xbee Shield akan tampak seperti pada **Gambar 3. 13**.



**Gambar 3. 13** Arduino + XBee + XBee Shield

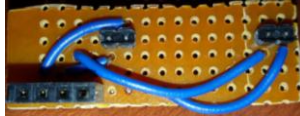
### 3.2.4 Konektor Modul *Real-Time Clock* DS1307 ke Arduino

Bentuk komunikasi data dari IC RTC adalah I2C yang merupakan kepanjangan dari *Inter-Integrated Circuit*. Komunikasi jenis ini hanya menggunakan 2 jalur komunikasi, yaitu SCL dan SDA. Semua mikrokontroler sudah dilengkapi dengan fitur komunikasi 2 jalur ini, termasuk diantaranya Mikrokontroler Arduino yang diberikan pada **Gambar 3. 14**.



**Gambar 3. 14** Rangkaian RTC dengan Mikrokontroler

Agar lebih memudahkan dalam penyambungannya atau koneksi, perlu dibuatnya suatu *port* konektor yang digunakan sebagai sambungan antara Arduino dengan modul RTC ini. *Port* konektor dapat disebut sebagai *RTCDS1307-to-Arduino Connector*. Hasil dari pembuatan, ditampilkan dengan hasil solderan pada **Gambar 3. 15** tampak atas dan **Gambar 3. 16** tampak bawah.



**Gambar 3. 15** *Port* Konektor Tampak Atas

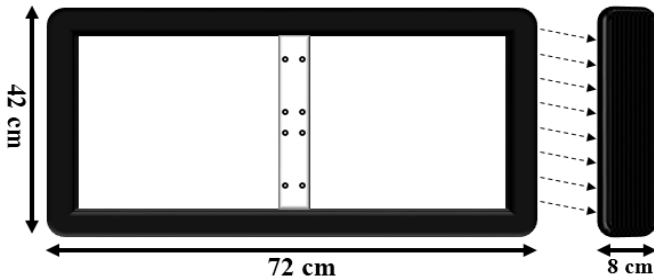


**Gambar 3. 16** *Port* Konektor Tampak Bawah

Komponen RTC DS1307 memiliki ketelitian dengan *error* sebesar 1 menit per tahunnya.

### 3.2.5 Kotak atau Bingkai *Multi-display*

Agar penampilan dari *Multi-display* menjadi lebih rapi dan mempunyai bentuk yang tetap, maka dirancanglah suatu kotak atau bingkai yang digunakan untuk menempatkan *Multi-display*. Bingkai akan dibuat dari bahas aluminium, karena lebih ringan dan fleksibel ketika ingin dibawa kemana saja. Desain dari rancangan pembuatan bingkai ini diberikan pada **Gambar 3. 17**.



**Gambar 3. 17** Desain Bingkai untuk *Multi-display*

Pada gambar di atas adalah desain bingkai untuk meletakkan panel P10, atau *Multi-display*, dengan ukuran yaitu panjang 72 cm, lebar 42 cm, dan tinggi 8 cm. Pada bagian sisi juga diberikan bagian yang diberi nama siku untuk memperkuat dari desain ini. Bagian tengah dari desain ini diberikan sebuah *din rail* sebagai penopang dari modul-modul DMD P10 yang akan dipasang. Realisasi dari desain tersebut ditunjukkan pada **Gambar 3. 18**.



**Gambar 3. 18** Hasil dari Rancangan Bingkai *Multi-display*

### 3.3 Perancangan Perangkat Lunak (*Software*)

Dalam subbab ini, akan dibahas mengenai perancangan perangkat lunak yang mendukung seluruh perancangan dan pembuatan alat yang menggunakan *software* IDE Arduino 1.8.0.

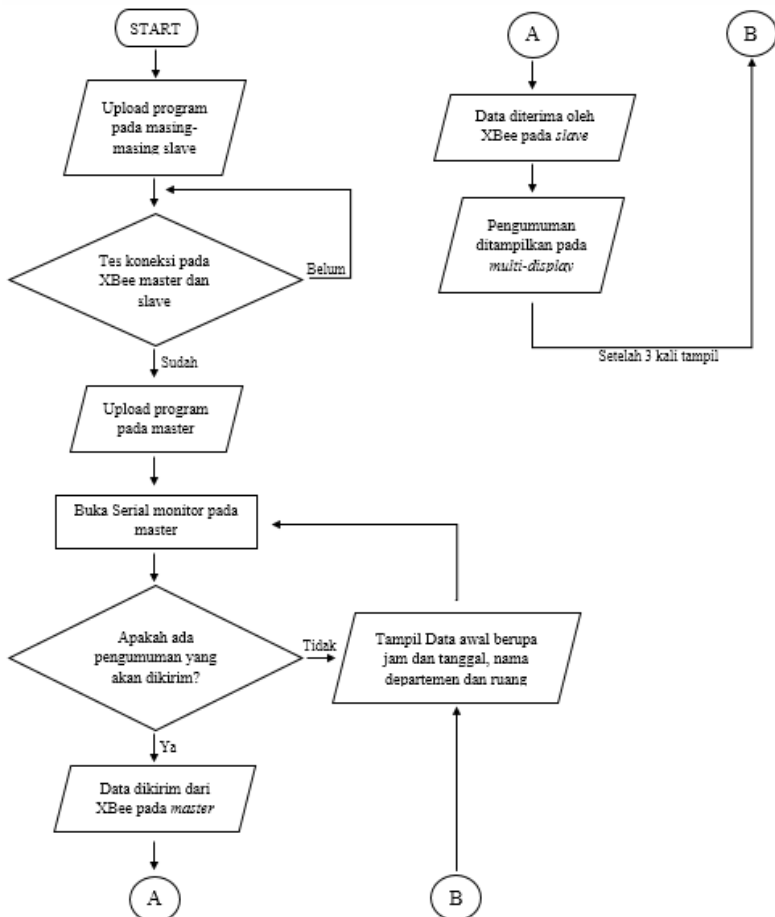
Setelah semua *hardware* terangkai dan terintegrasi serta telah diuji kinerjanya maka dilakukan pembuatan program keseluruhan. Alur dari program secara keseluruhan ini dapat dilihat pada *flowchart* **Gambar 3. 19**. Pada gambar tersebut, hal pertama yang harus dilakukan adalah membuat dan melakukan *upload*-ing program yang berisi data khusus pada masing-masing *Slave* agar tersimpan pada penyimpanan atau memori Arduino melalui PC (*Personal Computer*). Ini dimaksudkan, agar saat nanti menggunakan *power supply switching*, Arduino langsung menampilkan data yang ada pada penyimpanannya ke *display Multi-display*. Selanjutnya, dilakukan pengamatan apakah antara XBee pada *master* telah terkoneksi dengan XBee yang ada pada tiap *Slave*. Ini ditandai dengan berkedipnya lampu / *LED* indikator berwarna merah yang ada pada *shield* XBee. Jika telah terkoneksi, dilakukan *upload*-ing program khusus yang ada pada *master* untuk melakukan komunikasi nirkabel / *Wireless* antara *master* dengan *Slave*. Setelah itu, *operator* pada *master* membuka *serial monitor* yang ada pada Arduino untuk

menuliskan apa yang ingin diumumkan. Jika tidak ada yang ingin diumumkan, maka *Multi-display* akan menampilkan data berupa jam dan tanggal, nama departemen, dan nama ruang. Jika ada, maka *Multi-display* akan menampilkan pengumuman itu sebanyak 3 kali tampilan, lalu akan kembali menampilkan data jam dan tanggal, nama departemen, dan nama ruang.

Program data yang berupa nama departemen dan nama ruang tidak dapat diubah oleh *operator* yang berada di *master* ketika *Multi-display* telah terpasang. Program dapat diubah hanya jika operator meng-*upload* ulang program yang ada pada *Slave* atau dalam kasus *flowchart* ini, mengulangi dari awal proses. Dalam hal pengecekan apakah XBee pada *master* telah terkoneksi dengan XBee pada *Slave*, ketika *Multi-display* telah terpasang, dapat digunakan tolok ukur waktu hasil *trial-and-error* selama 10 detik ketika XBee *master* dan *Slave* telah terhubung ke catu daya masing-masing.

Ketika telah memasang XBee dan menunggu selama selang waktu yang telah ditentukan, seperti yang dijelaskan diatas yaitu selama 10 detik, pengguna atau *operator* lalu membuka *serial monitor* dan mencari *port COM* dari salah satu papan Arduino yang terpasang. Jika sudah, maka pengguna menuliskan apa yang ingin disampaikan lalu mengklik tombol '*send*' atau menekan tombol *Enter* yang tersedia pada *keyboard*. Maka, jika memakai 2 *Slave Multi-display*, pengumuman tersebut akan dikirimkan ke XBee *Slave* yang kemudian akan diterima oleh Arduino lalu menampilkan pengumuman tersebut sama antara *Slave* 1 dengan yang lain dikarenakan sinyal pengiriman XBee hanya bisa dikirimkan 1 arah atau *point-to-point*.



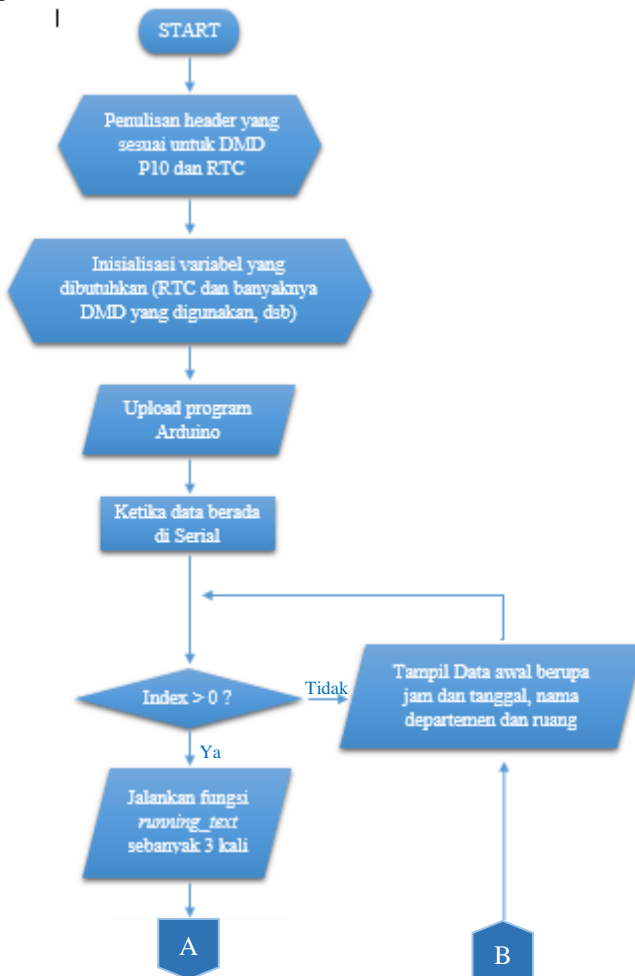


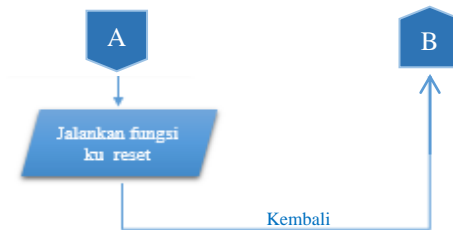
**Gambar 3. 19** Flowchart Keseluruhan Sistem

Pada tahap ini, akan dijelaskan mengenai perencanaan dan pembuatan dari perangkat lunak Pembuatan Perangkat *Multi-display* yang Dikontrol Secara Terpusat Menggunakan *Wireless* pada *Slave*-nya sebagai berikut.

### 3.3.1 Perancangan Program pada *Slave Multi-display*

Pada Pembuatan Perangkat Multi-display yang Dikontrol Secara Terpusat Menggunakan Wireless, dirancanglah suatu program untuk dapat menampilkan dari *software* IDE, yang menjadi jembatan dari Arduino, kepada tampilan *Multi-display*, terdiri dari DMD P10, XBee, dan Modul RTC DS1307. Pada DMD, dibutuhkan *library* *DMD.h* dan *SPI.h* agar Arduino dapat berkomunikasi dengan DMD lewat komunikasi SPI (*Serial Peripheral Interface*). Untuk modul RTC, diperlukan *library* *RTCLib.h*. Flowchart untuk program ini ditunjukkan pada **Gambar 3. 20**.



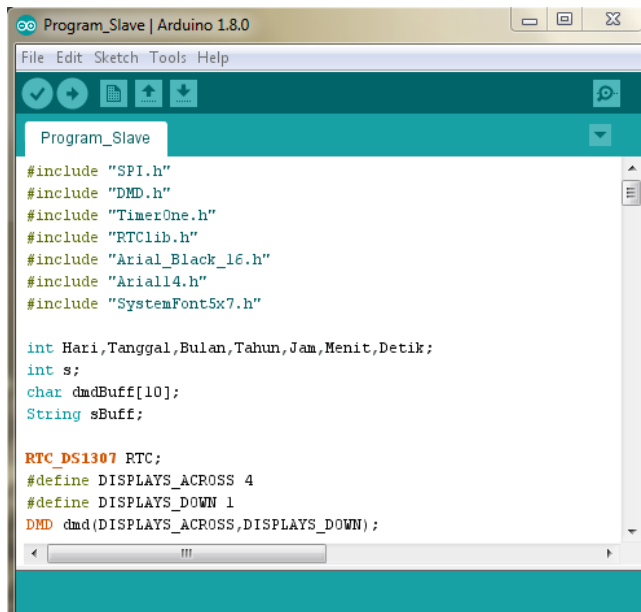


**Gambar 3. 20** Flowchart Program Pada Slave

Pada **Gambar 3. 20** dapat dilihat perancangan *flowchart* untuk program yang ada pada *Slave*. Urutan cara kerja *flowchart* adalah sebagai berikut.

- a. *Start* adalah ketika program dimulai.
- b. Penulisan *header* yang tepat untuk DMD P10 dan RTC agar program dapat diproses oleh modul-modul tersebut. Lalu menginisialisasi variabel yang dibutuhkan, seperti pada DMD diperlukan untuk menginisialisasi jumlah DMD yang digunakan berupa baris dan kolom (yang digunakan dalam tugas ini adalah 4 baris 1 kolom), dan pada RTC seperti memperkenalkan 'Tanggal', 'Bulan', 'Jam', 'Menit', dan lain-lain untuk keperluan selanjutnya dalam pemrograman.
- c. Mengupload program tersebut ke *Slave Multi-display* melalui kabel serial.
- d. Ketika data berada pada komunikasi serial, jika *index\** memiliki kondisi lebih dari 0 (nol), maka menjalankan fungsi *running\_text* yang nantinya digunakan sebagai fungsi untuk menyampaikan pengumuman berupa teks secara berjalan. Jika tidak, maka *Slave* akan menampilkan tampilan *standby* berupa tampilan Waktu dan Tanggal, nama departemen dan ruang yang diprogram dengan teks *static* atau tidak berjalan.  
 \* : *index* adalah *string* yang diinisialisasi dalam bentuk *byte* sebagai tempat penyimpanan karakter.
- e. Ketika fungsi *running\_text* berjalan, fungsi tersebut akan berulang sebanyak 3 kali. Setelah 3 kali, maka secara otomatis akan mengaktifkan fungsi *ku\_reset* yang berfungsi untuk mereset Arduino dan mengembalikannya ke keadaan tampilan *standby*.

**Gambar 3. 21** merupakan gambar *software* program IDE yang ada pada *Slave Multi-display*.



**Gambar 3. 21** Program Pada *Slave*

Dari program diatas, telah jelas bahwa yang digunakan adalah *dot matrix display* (DMD) dan *real-time clock* (RTC) yang dihubungkan dengan Arduino. Pada program tersebut juga terdapat banyak pembuatan fungsi baru dari *void* agar penulisannya menjadi lebih ringkas dan mudah untuk dianalisa. Fungsi-fungsi tersebut adalah *ku\_reset*, *tulis\_teks*, *cancel\_teks*, *running\_teks*, *TampilWaktu* dan *dept\_name*.

## BAB IV

### HASIL IMPLEMENTASI ALAT

Untuk mengetahui apakah tujuan-tujuan dari pembuatan alat ini telah sesuai dengan yang diharapkan atau tidak, maka dilakukan pengujian dan analisa terhadap alat yang telah dibuat.

Alat yang telah dibuat yaitu Pembuatan Perangkat *Multi-display* yang Dikontrol Secara Terpusat Menggunakan *Wireless*. Cara kerja dari alat tersebut yaitu pengguna yang berada di bagian *master*, bersama dengan Arduino dan XBee *coordinator*, dapat mengirimkan suatu pengumuman kepada beberapa *Multi-display* yang telah terpasang pada tempatnya masing-masing melalui *serial monitor* yang ada pada *software IDE*, lalu menampilkan pengumuman tersebut pada tampilan *Multi-display*. Hasil dari perancangan *Multi-display* ditunjukkan pada **Gambar 4. 1**.



**Gambar 4. 1** Hasil Perancangan *Multi-display*

Pada penjelasan diatas adalah cara kerja alat secara umum atau secara keseluruhan sistem jika telah menggunakan komunikasi *Wireless*. Tugas yang didapatkan penulis tidak sampai pada komunikasi *Wireless*, tetapi masih menggunakan komunikasi dengan kabel USB yang terhubung dengan PC atau laptop, jadi ketika tidak menggunakan XBee pun tidak masalah. Cara kerjanya tidak jauh berbeda dengan cara kerja secara menyeluruh, hanya saja disini pengirimannya langsung lewat PC yang terhubung langsung dengan Arduino pada *Multi-display* dengan kabel USB. Jadi, data langsung dikirim melalui *serial monitor* untuk selanjutnya ditampilkan pada tampilan *Multi-display*.

Pada saat menjalankan cara kerja yang hanya menggunakan komunikasi serial ini, pada *serial monitor*, bisa dilakukan pengamatan

atau *monitoring* apakah data yang dikirim sudah tampil atau belum pada *Multi-display*. Data telah terkirim dan akan tampil ketika data yang dimasukkan akan tertulis ulang pada tampilan *serial monitor*.

#### 4.1 Pengujian *Multi-display*

Pada subbab ini akan dijelaskan mengenai hasil pengukuran tegangan pada tiap-tiap komponen, dan pengujian pengiriman data dari hasil perancangan alat seperti yang telah di bahas pada **BAB 3**. Tujuan pengujian alat ini adalah untuk mengetahui ketepatan dan ketelitian dari alat yang dibuat, sehingga dengan melakukan pengujian-pengujian secara bertahap pada *Multi-display* akan diketahui kekurangan-kekurangan yang mungkin bisa untuk lebih diperbaiki lagi kedepannya. Dimana dari dua *Slave Multi-display* yang dibuat, data yang diambil hanya pada satu salah satu *Slave* saja, dikarenakan kesamaan perancangan dan pemrogramannya.

##### 4.1.1 Pengujian Tegangan pada *Multi-display*

Pengujian tegangan pada *Multi-display* dilakukan pada tiap modul DMD P10 yang digunakan, *power supply* dan pada kontroler Arduino. Pengukuran tegangan pada komponen-komponen tersebut dilakukan untuk mencari apakah tegangan yang diukur sesuai atau sama dengan yang ada pada teori. Pada teorinya, tegangan pengoperasian yang dibutuhkan oleh 1 modul DMD P10 dan Arduino adalah sebesar 5V. Pada *power supply*, karena telah tertulis pada segel, tegangan yang dikeluarkan adalah sebesar 5V. Penyambungan untuk sumber tegangan dari *power supply* ke modul-modul DMD P10 dan kontroler Arduino ini dilakukan secara paralel.

Jika saat diukur, terjadi perbedaan tegangan dengan yang ada pada teori maka dapat diketahui perhitungan *error* yang terjadi dalam persentase atau disebut juga persentase kesalahan. Rumus yang digunakan untuk menghitung persentase kesalahan ini diberikan dibawah ini.

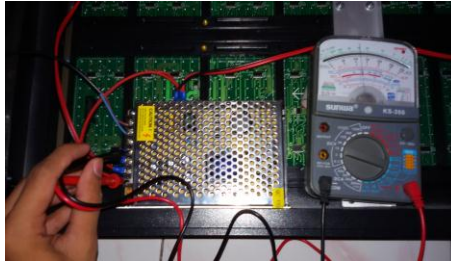
$$\text{Persentase Kesalahan} = \frac{V_M - V_T}{V_T} \times 100\%$$

Keterangan :

$V_M$  = Tegangan hasil pengukuran

$V_T$  = Tegangan sesuai teori

Pada **Gambar 4. 2** ditunjukkan kondisi pada saat mengukur tegangan keluaran (*output*) yang ada pada *power supply*.



**Gambar 4. 2** Pengukuran Tegangan *Output* pada *Power Supply*

Hasil pengukuran yang didapatkan adalah sebesar 5V. Jika dimasukkan dalam perhitungan maka akan menghasilkan:

$$\text{Persentase Kesalahan} = \frac{V_M - V_T}{V_T} \times 100\%$$

$$\text{Persentase Kesalahan} = \frac{5 - 5}{5} \times 100\%$$

$$\text{Persentase Kesalahan} = \frac{0}{5} \times 100\%$$

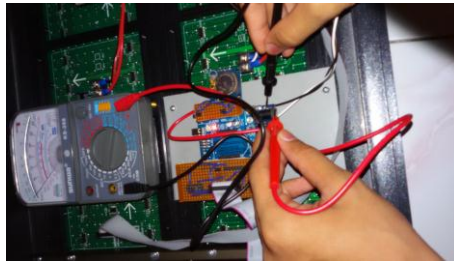
$$\text{Persentase Kesalahan} = 0\%$$

Dari hasil perhitungan, didapatkan hasil bahwa persentase kesalahan pengukuran adalah sebesar 0% yang menandakan hasil pengukuran sesuai dengan yang ada pada teori sehingga tidak terjadi kesalahan pengukuran.

Pada **Gambar 4. 3** dan **Gambar 4. 4** secara berurutan ditunjukkan kondisi pada saat pengukuran tegangan masukkan (*input*) pada salah satu modul DMD P10 dan pada kontroler Arduino.



**Gambar 4. 3** Pengukuran Tegangan *Input* pada Modul DMD P10



**Gambar 4. 4** Pengukuran Tegangan *Input* pada Kontroler Arduino

Hasil pengukuran yang didapatkan adalah sama, yaitu menghasilkan tegangan yang terukur sebesar 4,9V. Ini terjadi dikarenakan pemasangan sumber tegangan dari *power supply* ke tiap-tiap modul DMD P10 dan ke kontroler Arduino adalah secara paralel. Diketahui secara umum bahwa jika suatu rangkaian berbeban dipasang secara paralel, maka akan menghasilkan tegangan yang sama pada tiap-tiap bebannya. Dari data tersebut, didapatkan hasil sebagai berikut.

$$\text{Persentase Kesalahan} = \frac{V_M - V_T}{V_T} \times 100\%$$

$$\text{Persentase Kesalahan} = \frac{4,9 - 5}{5} \times 100\%$$

$$\text{Persentase Kesalahan} = \frac{0,1}{5} \times 100\%$$

$$\text{Persentase Kesalahan} = 2\%$$



Dari hasil perhitungan, didapatkan hasil bahwa persentase kesalahan pengukuran adalah sebesar 2% yang menandakan bahwa hasil pengukuran belum sesuai dengan yang ada pada teori karena terjadi perbedaan sebesar 0,1V dari yang ada pada teori. Secara logika, *error* atau kesalahan sebesar 2% atau 0,1V dapat termasuk kedalam kategori *error* yang kecil, tetapi tetap saja tidak sesuai dengan yang ada pada teori yang ada karena terjadi penurunan tegangan. Jika dianalisis, faktor-faktor yang kemungkinan dapat menjadi penyebab penurunan tegangan ini adalah antara lain:

- Kabel yang digunakan sebagai penghubung antara sumber tegangan dan komponen yang akan dicatu memiliki resistansi yang menyebabkan rugi-rugi tenaga.
- Penggunaan tegangan yang terjadi pada rangkaian dalam dari modul DMD P10 yang terdiri dari berbagai IC dan 512 buah LED yang membutuhkan daya untuk menyalakannya.
- Penggunaan pada modul-modul yang terpasang pada Arduino, seperti XBee + *Shield*, konektor *DMDP10-to-Arduino*, dan modul RTC DS1307.

Pada **Tabel 4. 1** diberikan rangkuman dari hasil pengukuran agar lebih mudah dalam melihat dan menganalisisnya.

**Tabel 4. 1** Data Pengukuran Tegangan pada *Multi-display*

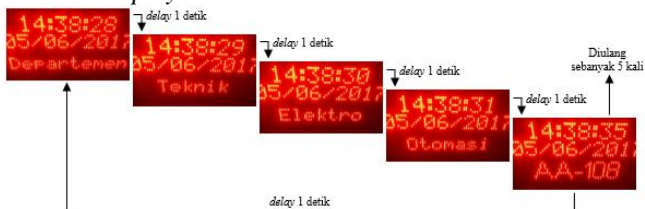
No.	Komponen	$V_T$ (Tegangan Secara Teoritis)	$V_M$ (Tegangan Hasil Pengukuran)	<i>Error</i> (Persentase Kesalahan)
1	<i>Power Supply</i>	5V	5V	0%
2	Modul DMD P10	5V	4,9V	2%
3	Kontroler Arduino	5V	4,9V	2%

#### 4.1.2 Pengujian Pengiriman Data

Pengujian pengiriman data ini dilakukan dengan cara komunikasi serial melalui kabel USB antara PC (*Personal Computer*) atau laptop dengan Arduino yang ada atau terpasang pada *Multi-display*. Data utama yang menjadi tujuan Tugas Akhir ini, yaitu suatu *running text* untuk menyampaikan suatu pengumuman akan dikirimkan melalui *serial*

monitor yang ada pada *software* IDE. Hasil dan pembahasannya akan dijelaskan sebagai berikut.

Setelah dilakukan *compiling* dan *uploading* program yang telah dibuat untuk *Slave Multi-display*, maka akan menghasilkan tampilan pada *Multi-display* berupa tampilan *standby*-nya yaitu menampilkan waktu dan tanggal, nama departemen dan nama ruang. Dibawah ini diberikan **Gambar 4. 5** yang mengilustrasikan penampilan pada tampilan *Multi-display*.



**Gambar 4. 5** Hasil Ilustrasi Tampilan *Standby* pada *Multi-display*

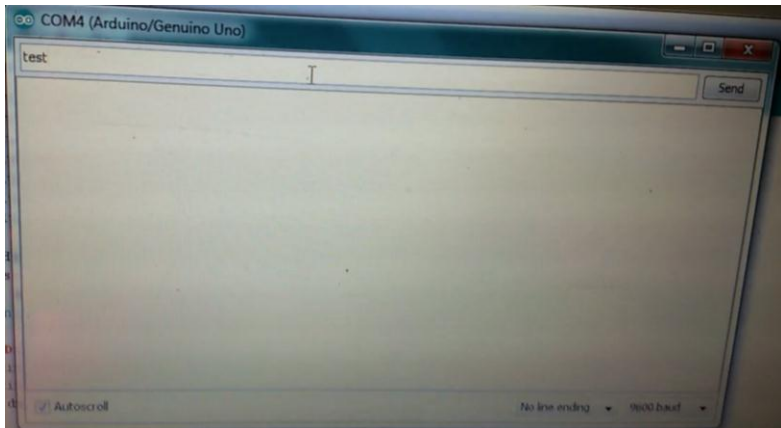
Pada gambar diatas, didapatkan hasil dari tampilan *standby Multi-display* yang berupa waktu dan tanggal, nama departemen dan ruang. Pada pemrogramannya, tiap 1 gambar diatas memiliki program masing-masing, ada bagian yang sama dan dibedakan, yang dipisahkan oleh *delay* sebesar 1 detik. Bagian yang sama adalah pada pemrograman waktu dan tanggal-nya yang diwakilkan dengan fungsi *TampilWaktu*, dan bagian yang berbeda adalah pada bagian pemrograman *static text* yang berada di bawah dari waktu dan tanggal, yaitu 'Departemen', 'Teknik', 'Elektro', 'Otomasi', 'AA-108'.

Penulisan tampilan nama departemen dan ruang dibuat berbentuk *static* dikarenakan agar terlihat jelas walaupun hanya akan tampil selama 1 detik ditiap tampilannya, dan juga untuk menyesuaikan dari gerakan detik dari penampil waktu dikarenakan pemrograman ini tidak dapat dipisahkan 1 sama lain. Tidak dapat dipisahkan karena adanya perintah dari *dmd.clearScreen(true)* dan *delay* pada tiap programnya. Jika tidak ada perintah tersebut, khususnya *delay*, maka waktu dan tanggal dari fungsi *TampilWaktu* tidak akan tampil atau hanya akan menyalakan beberapa LED secara acak yang tidak bisa terbaca.

Khusus pada bagian penulisan pemrograman tampilan ruang, akan diulang sebanyak 5 kali dikarenakan agar ruangan yang terpasang *Multi-display* didepannya dapat diketahui nama ruangnya. Perulangan pada bagian ini menggunakan fungsi *for*. Setelah berulang sebanyak 5 kali, maka tampilannya akan kembali menampilkan tampilan waktu dan

tanggal dan nama departemen ‘Departemen’, dan begitu seterusnya jika tidak ada pengumuman yang ingin disampaikan.

Jika ada pengumuman yang ingin disampaikan, maka bagian pemrograman yang menampilkan ruang setelah berulang sebanyak 5 kali, maka di saat perulangan yang kelima kali akan berhenti sejenak untuk kemudian menampilkan pengumuman yang ingin disampaikan. Pengumuman ini dikirimkan dari PC atau laptop melalui *serial monitor* yang ada pada *software* IDE. Lama waktu perhentian dari program menampilkan nama ruang yang dibutuhkan untuk menampilkan pengumuman tergantung dari banyaknya karakter yang dimasukkan. Berdasarkan hasil *trial-and-error* yang dicoba oleh penulis, lamanya berkisar antara 3 sampai 15 detik untuk pengirimannya. Pada **Gambar 4. 6**, diberikan contoh pengiriman suatu pengumuman melalui *serial monitor*.



**Gambar 4. 6** Pengiriman Pengumuman Melalui *Serial Monitor*

Pada gambar diatas, diberikan contoh mengirimkan pengumuman yang bertuliskan “*test*” untuk ditampilkan pada *Multi-display*. Hasil yang didapatkan ditunjukkan pada **Gambar 4. 7**.



**Gambar 4. 7** Hasil Tampilan *Multi-display*

Lamanya waktu yang dibutuhkan untuk mengirimkan pengumuman ini berkisar antara 0,5 sampai 1,8 detik saja dikarenakan karakter yang dikirimkan berjumlah 4 karakter. Ketika dicoba memasukkan data berupa jumlah karakter maksimum yang dapat dikirimkan oleh program yang ada, maka lama waktunya pun semakin lama dikarenakan pengiriman data yang dilakukan secara serial. Sebagai catatan, dalam penulisan pengumuman ini terdapat jumlah karakter maksimum yang dapat dikirimkan, yaitu sebanyak 63 karakter. Untuk lebih jelas dalam proses pengiriman data dari PC/Laptop ke tampilan *Multi-display*, dapat dilihat pada **Lampiran A.5**.

Jika dianalisis, terdapat hubungannya dengan penggunaan tipe data *byte* yang ada pada inisialisasi awal yang digunakan untuk menyimpan karakter. Tipe data *byte* memiliki 8 *bits data*, yang memiliki *range* 0 – 255. Tetapi ketika penulis mengganti tipe data menjadi yang lain, seperti *word*, tetap saja menghasilkan karakter maksimal sebanyak 63 karakter. Juga ketika mencoba menganalisis bagian-bagian program yang lain dengan metode *trial-and-error*, belum juga mendapatkan solusi untuk dapat menambahkan jumlah karakter. Jika diperhitungkan secara matematis, maka didapatkan hasil perhitungan:

$$X = \frac{RM}{N}$$

$$\begin{aligned}
 &= \frac{256}{63 \text{ karakter}} \\
 &= 4,063492063/\text{karakter} \\
 &= 4/\text{karakter}
 \end{aligned}$$

Keterangan:

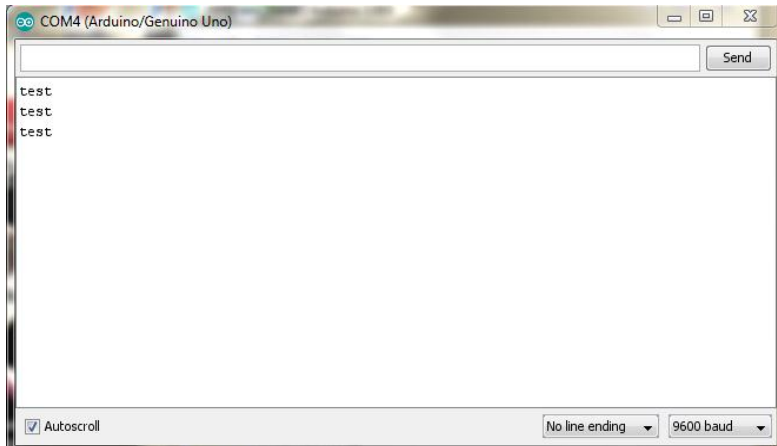
X : Jumlah tipe data yang mewakili 1 karakter

RM : *Range* dari tipe data, dalam hal ini *byte*

N : Jumlah karakter yang dapat ditampilkan oleh *Multi-display*

Dari hasil perhitungan tersebut, didapatkan hasil perhitungan sebesar 4,063492063/karakter atau jika dibulatkan menjadi 4/karakter yang berarti 1 karakter pada *Multi-display* mewakili sebanyak 4 satuan tipe data (*byte*). Jika diganti dengan tipe data *word* yang memiliki 16 *bits data*, maka hasil perhitungan tadi dikalikan 2 sesuai dengan *bits data* yang dimiliki oleh tipe data tersebut.

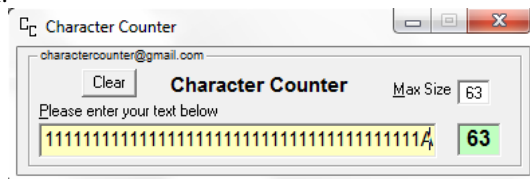
Data yang dikirimkan ini, yang kemudian ditampilkan pada *Multi-display*, dapat dilakukan *monitoring* pada *Slave Multi-display* melalui *serial monitor*. *Monitoring* ini bertujuan untuk mengamati apakah data yang dikirim telah diterima oleh *Slave* secara lengkap tanpa ada yang lewat atau hilang saat pengiriman. Selama pengujian yang dilakukan oleh penulis, data yang dikirimkan diterima dan ditampilkan secara lengkap pada tampilan *Multi-display*. Syarat yang harus dipenuhi adalah Arduino pada *Multi-display* masih terhubung atau terkoneksi melalui kabel USB dengan PC atau laptop.

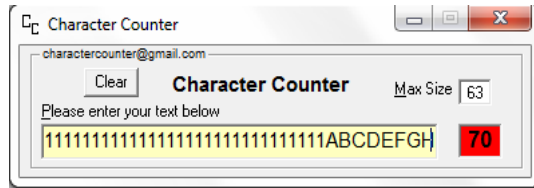


**Gambar 4. 8** Monitoring pada Slave Melalui Serial Monitor

Pada **Gambar 4. 8**, didapatkan hasil *monitoring* tampilan *Slave* yang bertuliskan ‘test’, yang telah berulang sebanyak 3 kali dan ditampilkan secara lengkap tanpa ada yang lewat atau hilang dari karakter atau data yang dikirimkan. Jika ingin memasukkan data baru, tinggal menuliskannya pada *box* yang telah disediakan oleh *serial monitor*, dan harus menunggu pengumuman yang sebelumnya telah selesai berulang sebanyak 3 kali atau telah kembali kepada tampilan *standby*-nya. Jika tidak, maka data yang baru tidak akan ditampilkan.

Dalam memudahkan dalam penghitungan jumlah karakter yang akan dimasukkan ini, penulis menggunakan aplikasi *open source* dari internet yang bernama *Character Counter V6*. Aplikasi ini digunakan untuk menghitung jumlah karakter yang dituliskan padanya, spasi juga termasuk dalam perhitungan. **Gambar 4. 9** menunjukkan penggunaan aplikasi ini.

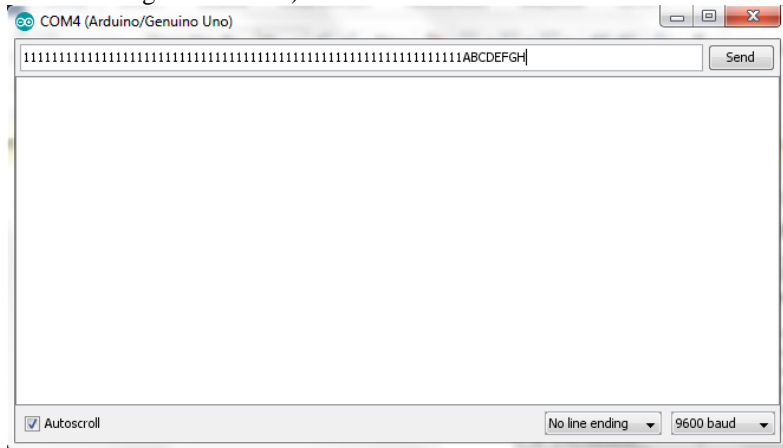




**Gambar 4. 9** Penggunaan Aplikasi *Character Counter V6*

Pada gambar diatas, diketahui 2 kondisi yaitu pada saat karakter yang dimasukkan tepat pada maksimal karakter, kotak *counter*-nya tetap berwarna hijau, dengan yang melebihi maksimal karakter yang ditentukan, kotak *counter*-nya berubah warna menjadi merah. Untuk maksimal karakter-nya, pengguna dapat menuliskan sendiri sesuai keinginan, tetapi dikarenakan pada tugas maksimal hanya 63 karakter, maka digunakanlah batas tersebut.

Dari jumlah karakter yang telah didapatkan tersebut, maka diujicobakan dikirimkan melalui *serial monitor* yang terdapat pada **Gambar 4. 10** (disini dipakai jumlah karakter sebanyak 70 karakter berdasarkan gambar diatas).



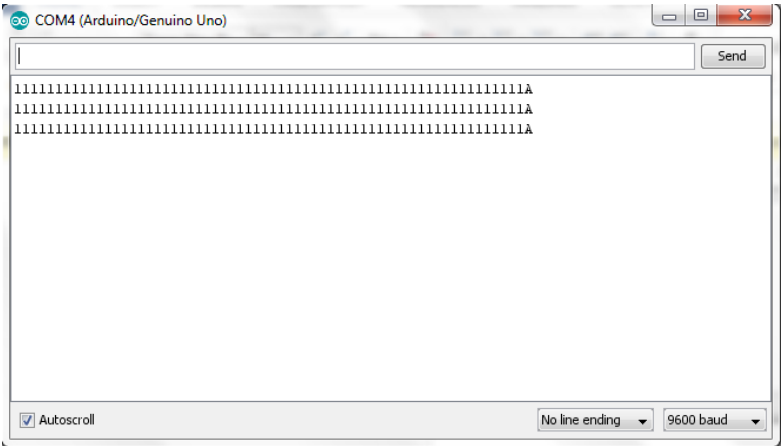
**Gambar 4. 10** Menuliskan Data Sebanyak 70 Karakter

Walaupun yang dikirimkan sebanyak 70 karakter, tetap saja yang akan di-*monitor* dan di tampilkan pada *Multi-display* sebanyak 63 karakter, seperti yang ditunjukkan pada **Gambar 4. 11**. 7 karakter lainnya tidak terkirim dikarenakan sudah melewati dari *range* yang ada.

Hasil *monitoring* pada seperti yang ditunjukkan oleh **Gambar 4. 12** hanya menampilkan karakter sebanyak 63 karakter.



**Gambar 4. 11** Hasil Tampilan *Multi-display*



**Gambar 4. 12** Hasil *Monitoring* pada *Serial Monitor*

Hasil pengujian pada **Tabel 4. 2** menunjukkan perbandingan dari hasil data yang dikirimkan pada PC/Laptop dengan hasil yang ditampilkan pada perangkat *Multi-display* sebagai berikut.



**Tabel 4. 2** Hasil Pengujian Pengiriman Data

No.	Data yang dikirimkan dari Laptop/PC	Data yang diterima dan ditampilkan oleh <i>Multi-display</i>
1	Test	
2	Coba	
3	Percobaan	
4	Selamat Datang!	
5	Berkumpul Sekarang	
6	Buka Mata, Buka Hati	

7	Harap Tenang! Sedang Ujian!	
8	Pengumpulan Persyaratan Yudisium	
9	Pada Hari Jumat, 28 Juli 2017 di	
10	Sekretariat Departemen Teknik Elektro Otomasi, Fakultas Vokasi-ITS	

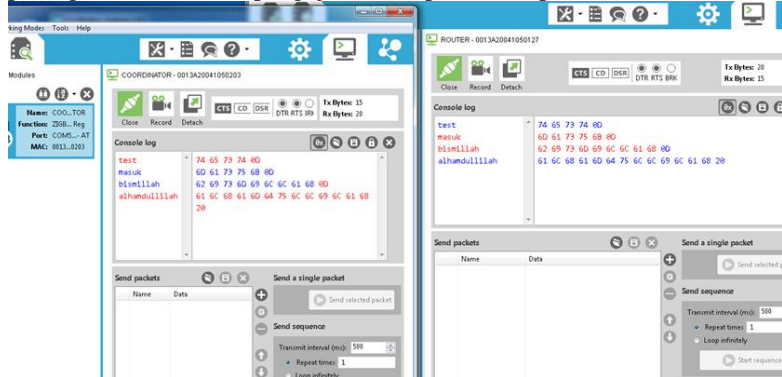
Dari hasil data pada tabel diatas, dapat diketahui bahwa seluruh data yang dikirimkan, 10 data, dari PC/Laptop dapat diterima dan ditampilkan oleh *Multi-display* dengan hasil yang sama. Hanya saja, pada pengiriman data yang terakhir, data ke -10, terjadi hilangnya 3 karakter data yang bertuliskan 'ITS'. Ini terjadi karena perangkat ini memiliki jumlah karakter maksimum yang harus ditaati, yaitu sebanyak 63 karakter, sedangkan karakter yang dikirimkan berjumlah 66 karakter, sehingga kata 'ITS' tidak ditampilkan pada tampilan *Multi-display*.

## 4.2 Pengujian Keseluruhan Sistem

Pada subbab ini, akan dipaparkan mengenai pengujian dari keseluruhan sistem yang digunakan dalam Tugas Akhir ini. Pengujian ini didasarkan pada diagram blok keseluruhan sistem yang terdapat pada **BAB III** di atas yang menjelaskan mengenai hubungan perangkat *Master* dengan perangkat *Slave* melalui komunikasi *wireless* XBee yang ada pada kedua perangkat tersebut.

### 4.2.1 Pengujian XBee Pada Program XCTU

Pengujian ini dilakukan dengan menggunakan program yang bernama XCTU. Program XCTU digunakan untuk mengalamatkan XBee menjadi *coordinator* dan *router* dimana untuk *coordinator* akan ditempatkan pada perangkat *Master* dan *router* akan ditempatkan pada perangkat *Slave*. Hasil pengujian ini dapat dilihat pada **Gambar 4. 13**.



**Gambar 4. 13** Pengujian dengan Program XCTU

Pada gambar tersebut merupakan hasil pengujian menggunakan Program XCTU antara XBee *coordinator* dan *router*. Data yang dikirimkan oleh XBee *coordinator* diterima sepenuhnya oleh XBee *router* sehingga pada tampilan program XCTU menghasilkan data yang sama, yang berarti bahwa XBee tersebut sudah berfungsi dengan baik dan siap digunakan sebagai modul komunikasi antara perangkat *Master* dan *Slave*. **Tabel 4. 3** memberikan hasil pengujian dari pengiriman data antar XBee pada program XCTU.

**Tabel 4. 3** Pengujian Pengiriman Data Dengan Program XCTU

<i>Coordinator</i>	Code	<i>Router</i>	Code
TEST	74 65 73 74 OD	TEST	74 65 73 74 OD
MASUK	6D 61 73 75 6D	MASUK	6D 61 73 75 6D
ALHAMDULIL LAH	61 6C 68 61 6D 64 75 6C 69 6C 6C 61 68	ALHAMDULIL LAH	61 6C 68 61 6D 64 75 6C 69 6C 6C 61 68
DONE	64 6F 6E 65	DONE	64 6F 6E 65

#### 4.2.2 Banyaknya Karakter yang Dapat Dikirimkan dan Kisaran Waktu dalam Menampilkannya

Setelah XBee diuji dan data yang dihasilkan sesuai, maka XBee dipasangkan pada perangkat *Master* dan *Slave*. Pada perangkat *Slave*, dilakukan *upload*-ing program untuk mengisi data dari Arduino. Lalu, apabila perangkat *Slave* ini mendapatkan catu daya, maka perangkat ini akan menampilkan tampilan *standby*, yaitu menampilkan jam dan tanggal, nama departemen serta nama ruang seperti yang ditunjukkan pada **Gambar 4. 5** diatas.

Dalam hal banyaknya karakter yang dapat dikirimkan, baik melalui sambungan antara Perangkat *Slave Multi-display* yang langsung terhubung dengan PC atau Laptop melalui kabel USB maupun melalui komunikasi *wireless* XBee yang ada di perangkat *Master* dan *Slave* jumlah karakter yang dapat dikirimkan maksimal hanya 63 karakter. **Gambar 4. 14** memperlihatkan perangkat *Master* dari Tugas Akhir ini. Pengujian yang dilakukan adalah menuliskan karakter sebanyak 70 karakter, namun yang dapat ditampilkan hanya 63 karakter (karakter yang dimasukkan sama dengan pengujian pada pembahasan sebelumnya; lihat pada **Tabel 4.2**). Pada **Gambar 4. 14**, menunjukkan perangkat *Master* yang sedang membuka bagian *serial monitor* Arduino untuk mengirimkan data pada perangkat *Slave*. Untuk lebih jelasnya apa yang ada pada gambar, lihat di bagian **Lampiran A.5**.



**Gambar 4. 14** Perangkat *Master*

Kisaran waktu dalam menampilkannya jika menggunakan komunikasi *wireless* XBee antara dua perangkat tersebut terdapat perbedaan waktu, namun tidak terlalu jauh berbeda. Jika perangkat *Slave* langsung terhubung dengan PC atau Laptop berkisar antara 0,5 – 1,8

detik, maka dengan komunikasi *wireless* XBee memerlukan waktu 1 – 2,2 detik. Untuk lebih jelasnya, lihat pada **Lampiran A.4**.

#### 4.2.3 Jarak yang Dapat Dijangkau oleh XBee S2C

Dalam pengujian jarak yang dapat dijangkau oleh XBee S2C, dilakukan 2 pengujian untuk yang tanpa halangan (LOS/*Line of Sight*) dan dengan halangan (dipisahkan ruang).

Pengujian tanpa halangan (LOS), dilakukan secara bertahap dengan pengiriman data, yaitu dari jarak 0 – 120 meter. Setelah data dikirimkan, dalam jangkauan jarak yang berkisar antara 0 – 100 meter, perangkat *Slave Multi-display* masih dapat menampilkan data yang dikirimkan, sedangkan selebihnya, data tidak ditampilkan oleh perangkat *Slave* tersebut. Sehingga, jarak maksimum yang dapat dijangkau secara *wireless* tanpa halangan adalah 100 meter. Pada **Tabel 4. 4** diberikan data dari hasil pengiriman data tanpa halangan.

Pengujian dengan halangan (dipisahkan ruang), juga dilakukan secara bertahap dengan pengiriman data, yaitu dari jarak 0 – 25 meter. Setelah data dikirimkan, dalam jangkauan jarak yang berkisar antara 0 – 20 meter, perangkat *Slave Multi-display* masih dapat menampilkan data yang dikirimkan, sedangkan selebihnya, data tidak ditampilkan oleh perangkat *Slave* tersebut. Sehingga, jarak maksimum yang dapat dijangkau secara *wireless* dengan halangan adalah 20 meter. **Gambar 4. 14** menunjukkan hasil tampilan *Multi-display* yang menerima data di dalam ruang yang berbeda. Pada **Tabel 4. 5** diberikan data dari hasil pengiriman data dengan halangan.



(a)



(b)

**Gambar 4. 15** Tampilan di Dalam Ruang, (a) *Slave 1*; (b) *Slave 2*

**Tabel 4. 4** Hasil Pengujian Pengiriman Data Tanpa Halangan

No.	Jarak	Status
1	0 meter	Terkirim
2	10 meter	Terkirim
3	20 meter	Terkirim

4	30 meter	Terkirim
5	40 meter	Terkirim
6	50 meter	Terkirim
7	60 meter	Terkirim
8	70 meter	Terkirim
9	80 meter	Terkirim
10	100 meter	Terkirim
11	110 meter	Tidak Terkirim
12	120 meter	Tidak Terkirim

**Tabel 4. 5** Hasil Pengujian Pengiriman Data Dengan Halangan

No.	Jarak	Status
1	0 meter	Terkirim
2	2 meter	Terkirim
3	5 meter	Terkirim
4	10 meter	Terkirim
5	15 meter	Terkirim
6	20 meter	Terkirim
7	25 meter	Tidak Terkirim

## **BAB V**

### **PENUTUP**

Setelah melakukan perencanaan, pembuatan, dan pengujian alat maka dapat diambil kesimpulan serta diberikan saran untuk pengembangan alat ini kedepannya.

#### **5.1 Kesimpulan**

Hasil dari perancangan serta implementasi alat dari Pembuatan Perangkat *Multi-display* yang Dikontrol Secara Terpusat Menggunakan *Wireless* dapat diambil kesimpulan sebagai berikut:

1. Dimensi dari *Multi-display* ini adalah 64x32 cm<sup>2</sup> dengan besar bingkai 72x42x8 cm<sup>3</sup>.
2. Tampilan awal *Multi-display* berupa jam dan tanggal, nama departemen, dan nama ruang atau *standby* tampil ketika *Multi-display* mendapatkan catu daya.
3. Pengumuman ditampilkan dengan menuliskannya pada *serial monitor* Arduino . Kisaran waktu untuk menampilkannya berkisar antara 0,5 – 1,8 detik dan melalui komunikasi *wireless* XBee berkisar antara 1 – 2,2 detik. Data tersebut akan tampil sebanyak 3 kali lalu kembali ke tampilan *standby*.
4. Jumlah karakter yang dapat dikirimkan dan diterima memiliki batas maksimal yang harus ditaati, yaitu sebanyak 63 karakter.
5. Berdasar hasil pengujian, data yang dikirimkan dari PC/Laptop dan data yang diterima dan ditampilkan oleh *Multi-display* adalah sama. Jika data yang dikirim melebihi 63 karakter, kelebihanannya tidak akan ditampilkan perangkat *Multi-display*.
6. Jarak yang dapat dijangkau dalam pengiriman data berupa pengumuman adalah untuk tanpa halangan (LOS/*Line of Sight*) sejauh 100 meter dan dengan halangan (dipisahkan ruang) sejauh 20 meter.
7. *Monitoring* pengumuman pada *Slave Multi-display* dapat dilakukan dengan menggunakan *serial monitor* Arduino setelah menuliskan pengumuman.

#### **5.2 Saran**

Untuk pengembangan dan penyempurnaan Pembuatan Perangkat *Multi-display* yang Dikontrol Secara Terpusat Menggunakan *Wireless*, maka diberikan beberapa saran sebagai berikut:

1. Memperkaya referensi bahan dalam pemrograman IDE Arduino yang berhubungan dengan modul DMD P10 dan modul RTC.
2. Dalam penelitian selanjutnya, diharapkan dapat mengirimkan data lebih dari 63 karakter.



## DAFTAR PUSTAKA

- [1] Anonim, 2013. *Mengenal dan Belajar Arduino UNO R3*. [Online]  
Available at: <http://ecadio.com/mengenal-dan-belajar-arduino-uno-r3>  
[Accessed 25 Mei 2017].
- [2] Anonim, 2014. *E-MOSFET (Enhancement-Metal-Oxide Semiconductor FET)*. [Online]  
Available at: <http://zonaelektro.net/e-mosfet-enhancement-metal-oxide-semiconductor-fet/>  
[Accessed 25 Mei 2017].
- [3] ANPEC, 2003. *APM4953 Dual P-Channel Enhancement Mode MOSFET*. 2 ed. s.l.:ANPEC Electronics Corp..
- [4] Arduino, S., 2017. *Mengenal Arduino Software (IDE)*. [Online]  
Available at: <http://www.sinauarduino.com/artikel/mengenal-arduino-software-ide/>  
[Accessed 1 Maret 2017].
- [5] Chandani Mauriya, Dr. M. B. Mali, "Wireless LED Matrix Display System, 2016 International Journal of Innovative Research in Computer and Communication Engineering, Pune, 2016, Vol. 4, Issue 6, "http://www.ijircce.com/upload/2016/june/193\_Wireless.pdf
- [6] Digi, 2016. *XBee/XBee-PRO S2C ZigBee RF Module User Guide*. 4th ed. s.l.:Digi International Inc..
- [7] Freetronics, 2016. *Using Your Freetronics DMD*. [Online]  
Available at: <https://www.freetronics.com.au/pages/using-your-freetronics-dmd#.WTbTuVV9600>  
[Accessed 25 Februari 2017].
- [8] Instruments, T., 2015. *SNx4HC04 Hex Inverters*. 3rd ed. s.l.:Texas Instruments Inc..
- [9] Lin, J., 2003. *Switching Power Supply Technical Manual*. s.l.:Mean Well Enterprises, Co. Ltd..
- [10] M. A. B. Sarijari, R. A. Rashid, M. R. A. Rahim, and N. H. Mahalin, "Wireless Home Security and Automation System Utilizing ZigBee based Multi-hop Communication," in

- Telecommunication Technologies 2008 and 2008 2nd Malaysia Conference on Photonics. NCTT-MCP 2008.6th National Conference on, 2008, pp. 242-245.
- [11] M. A. Kader, M. M. Rahman, S. M. I. B. Haider and M. Islam, "LED matrix based digital learning display for children with *Wireless* control," *2014 17th International Conference on Computer and Information Technology (ICCIT)*, Dhaka, 2014, pp. 397-400. doi: 10.1109/ICCITech.2014.7073134
- [12] Nexperia, 2016. *74HC\_HCT595*. 8th ed. s.l.:Nexperia B. V..
- [13] Pattnaik, A. A., 2016. *Wireless Notice Board (Arduino UNO + DMD)*. [Online] Available at: <https://create.arduino.cc/projecthub/abhijitbrain/Wireless-notice-board-arduino-uno-dmd-9ff29e> [Accessed 22 Maret 2017].
- [14] Philips, 2005. *74HC\_HCT245*. 3rd ed. s.l.:Koninklijke Philips Electronics N. V..
- [15] Robert Faludi. 2011 Building *Wireless* Sensor Networks - With Zigbee, Xbee, Arduino, And Processing - - O'Reilly –
- [16] Said, Pradana. 2010 Performance-Analysis-of-XBee-ZB-Module-Based-*Wireless*-Sensor-Networks
- [17] Semiconductors, N., 2015. *74HC\_HCT138*. 6th ed. s.l.:NXP Semiconductors N. V..
- [18] Kadir, A., 2014. *From Zero to A Pro Arduino*. 1st ed. Yogyakarta: ANDI.
- [19] Wibowo, S. E., 2015. *Pengetahuan Dasar RTC DS1307*. [Online] Available at: <https://proyekarduino.wordpress.com/2015/04/01/pengetahuan-dasar-rtc-ds1307/> [Accessed 25 Mei 2017].

## LAMPIRAN

### A.1. Dokumentasi Proses Masa Percobaan pada *Multi-display*



**Gambar 1.** Percobaan Menggunakan 1 Modul DMD P10



**Gambar 2.** Percobaan Menggunakan 2 Modul DMD P10



**Gambar 3.** Hasil yang Didapatkan dengan 2 Modul DMD P10



**Gambar 4.** Percobaan Menggunakan 4 Modul DMD P10

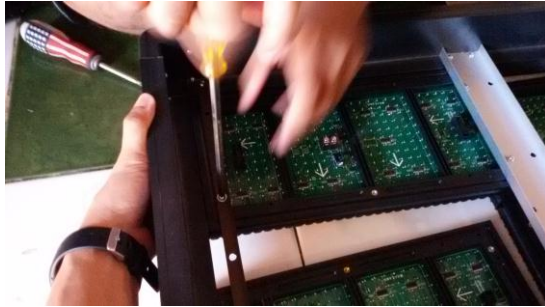


**Gambar 5.** Hasil yang Didapatkan dengan 4 Modul DMD P10

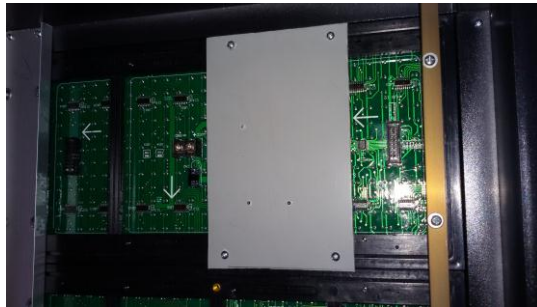
#### **A.2. Dokumentasi Proses Perakitan *Multi-display***



**Gambar 6.** Bingkai *Multi-display*



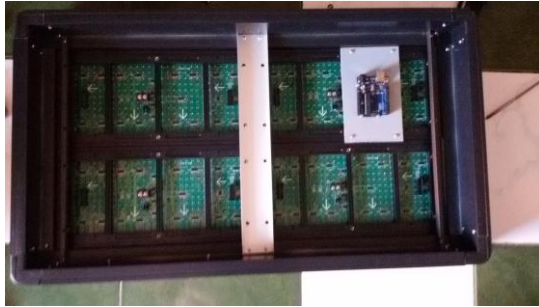
**Gambar 7.** Pemasangan Modul DMD P10 pada Bingkai



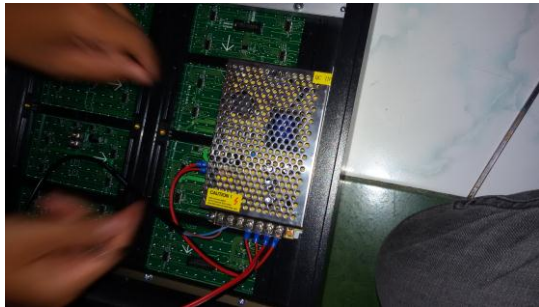
**Gambar 8.** Pemasangan Papan yang Digunakan untuk Menempatkan Kontroler Arduino



**Gambar 9.** Pemasangan Tahap Akhir dari Modul DMD P10



**Gambar 10.** Hasil yang Didapatkan Setelah Pemasangan



**Gambar 11.** Pemasangan *Power Supply Switching*



**Gambar 12.** Pemasangan Kabel *Power* pada Kontroler dan Modul DMD dari *Power Supply*

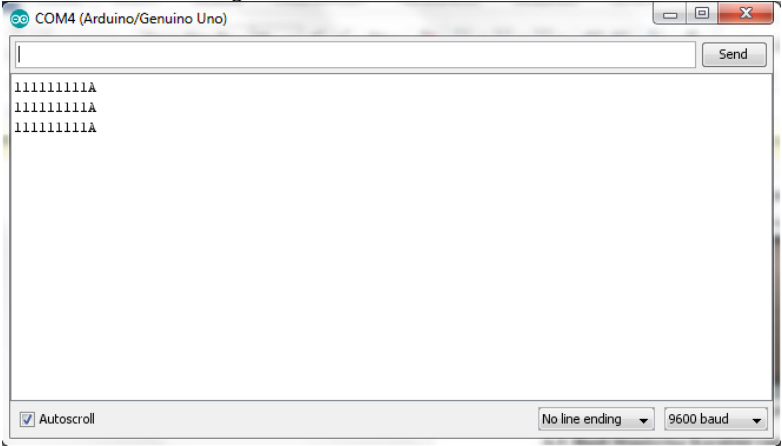


**Gambar 13.** Penyambungan Antara Modul DMD 1 dengan yang lain Secara Seri Menggunakan Kabel *Ribbon*

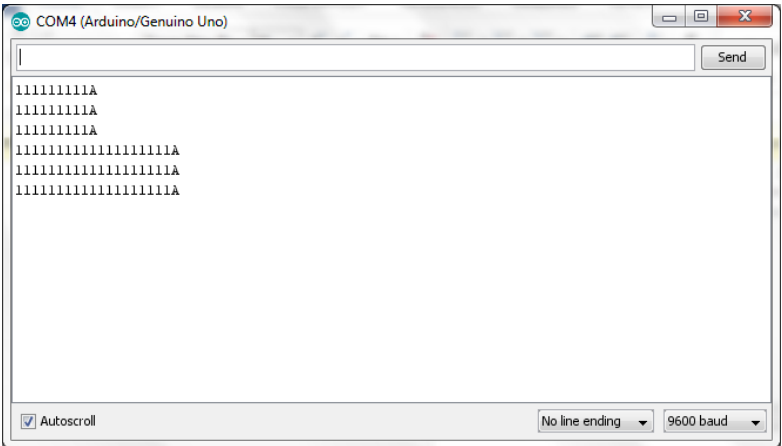


**Gambar 14.** Hasil Tampak Depan *Multi-display*

**A.3. Hasil *Monitoring* Karakter Melalui *Serial Monitor***

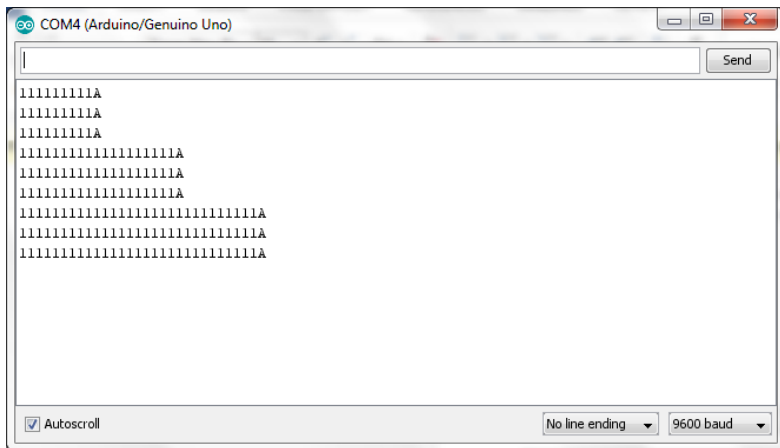


**Gambar 15.** Pengiriman 10 Karakter

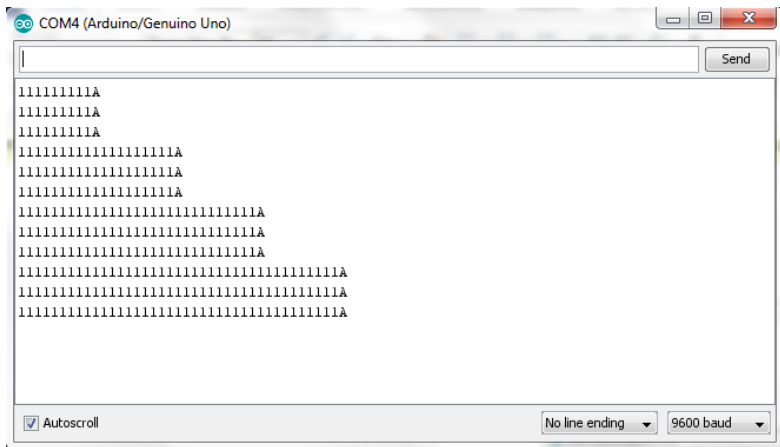


**Gambar 16.** Pengiriman 20 Karakter

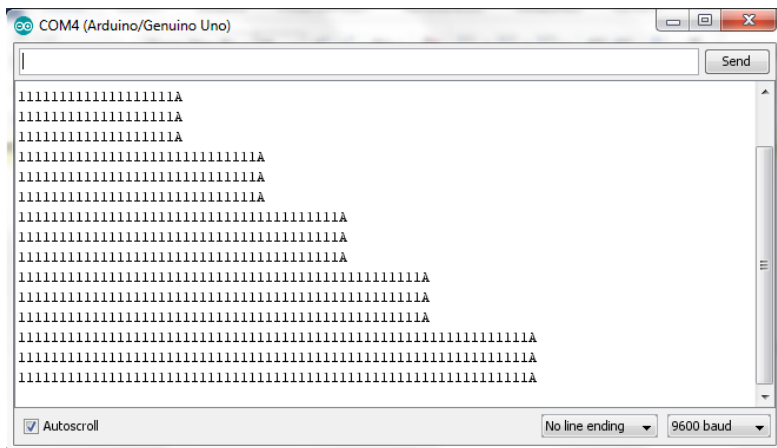
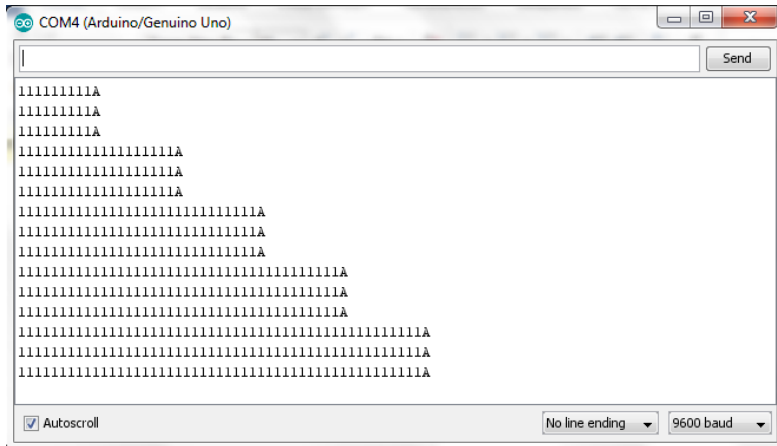




**Gambar 17.** Pengiriman 30 Karakter



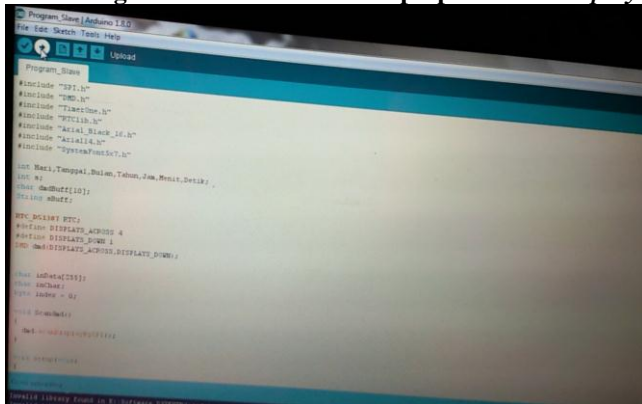
**Gambar 18.** Pengiriman 40 Karakter



#### A.4. Perhitungan Waktu Tunda Pengiriman Karakter

No.	Banyaknya Karakter yang Dikirimkan	Delay atau Penundaan Waktu saat	
		Pengiriman Data	Kembali ke Tampilan <i>Standby</i> (dari perulangan yang terakhir)
1	10 karakter	00'01"25	00'05"69
2	20 karakter	00'01"31	00'05"15
3	30 karakter	00'01"45	00'04"72
4	40 karakter	00'01"37	00'03"78
5	50 karakter	00'01"32	00'03"47
6	63 karakter	00'01"27	00'02"78

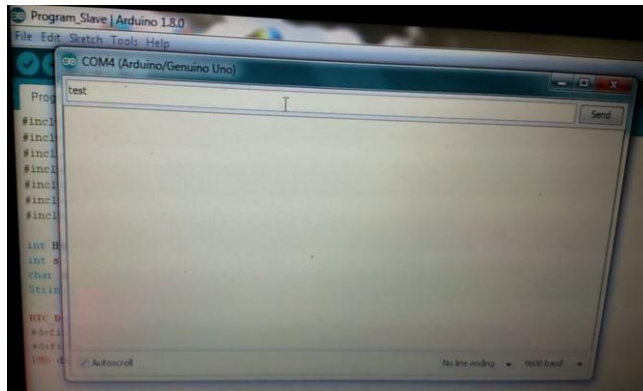
#### A.5. Proses Pengiriman Data dari PC/Laptop ke *Multi-display*



Gambar 21. *Compiling dan Uploading Program*



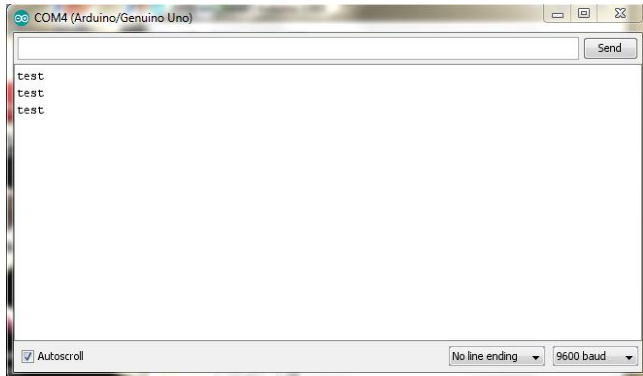
Gambar 22. *Ilustrasi Tampilan Standby pada Multi-display*



**Gambar 23.** Pengiriman Data Pengumuman Melalui *Serial Monitor*



**Gambar 24.** Pengumuman Ditampilkan pada *Multi-display*  
Sebanyak 3 kali



**Gambar 25.** Monitoring Data yang Dikirimkan dari *Serial Monitor* dan Ditampilkan oleh *Multi-display*

#### **A.6. Listing Program**

```
#include "SPI.h"
#include "DMD.h"
#include "TimerOne.h"
#include "RTCLib.h"
#include "Arial_Black_16.h"
#include "Arial14.h"
#include "SystemFont5x7.h"

int Hari,Tanggal,Bulan,Tahun,Jam,Menit,Detik;
int s;
char dmdBuff[10];
String sBuff;

RTC_DS1307 RTC;
#define DISPLAYS_ACROSS 4
#define DISPLAYS_DOWN 1
DMD dmd(DISPLAYS_ACROSS,DISPLAYS_DOWN);

#define max_char 100
char inData[max_char];
char inChar;
byte index = 0;
```

```

void Scandmd()
{
    dmd.scanDisplayBySPI();
}

void setup(void)
{
    RTC.begin();
    RTC.adjust(DateTime(F(__DATE__), F(__TIME__)));
    Serial.begin(9600);
    Timer1.initialize( 5000 );
    Timer1.attachInterrupt( Scandmd );
    dmd.clearScreen( true );
}

void(* ku_reset) (void) = 0;

void loop(void)
{
    while(Serial.available() == 0)
    {
        if (index > 0)
        {
            for(int i=0;i<3;i++)
            {
                running_text();
            }
            ku_reset();
        }else{
            dept_name();
            break;
        }
    }
    while(Serial.available() > 0)
    {
        dmd.clearScreen(true);
        if(index < max_char-1)
        {
            inChar = Serial.read();

```

```

        inData[index] = inChar;
        index++;
        inData[index] = 0;
        break;
    }
}
}

void running_text()
{
    Serial.println(inData);
    dmd.clearScreen(true);
    dmd.selectFont(Arial_14);
    dmd.drawMarquee(inData,max_char,(32*4)-1,1);
    long start=millis();
    long timer=start;
    boolean ret=false;
    while(!ret)
    {
        if ((timer+50) < millis())
        {
            ret=dmd.stepMarquee(-1,0);
            timer=millis();
        }
    }
}

void TampilWaktu()
{
    DateTime now = RTC.now();
    Tanggal = now.day();
    Bulan = now.month();
    Tahun = now.year();
    Jam = now.hour();
    Menit = now.minute();
    Detik = now.second();
    dmd.selectFont(System5x7);
    dmd.drawFilledBox( 23, 1, 24, 2, GRAPHICS_OR );
    dmd.drawFilledBox( 23, 4, 24, 5, GRAPHICS_OR );

```

```
dmd.drawFilledBox( 38, 1, 39, 2, GRAPHICS_OR );
dmd.drawFilledBox( 38, 4, 39, 5, GRAPHICS_OR );
dmd.drawString(15, 8, "/",1, GRAPHICS_OR);
dmd.drawString(33, 8, "/",1, GRAPHICS_OR);
```

```
// Display JAM
sBuff = Convert(now.hour());
s = sBuff.length()+1;
dmdBuff[s];
sBuff.toCharArray(dmdBuff,s);
dmd.selectFont(System5x7);
dmd.drawString(11,0,dmdBuff,s,0);
```

```
// Display Menit
sBuff = Convert(now.minute());
s = sBuff.length()+1;
dmdBuff[s];
sBuff.toCharArray(dmdBuff,s);
dmd.selectFont(System5x7);
dmd.drawString(26,0,dmdBuff,s,0);
```

```
// Display Detik
sBuff = Convert(now.second());
s = sBuff.length()+1;
dmdBuff[s];
sBuff.toCharArray(dmdBuff,s);
dmd.selectFont(System5x7);
dmd.drawString(41,0,dmdBuff,s,0);
```

```
//Display Tanggal
sBuff = Convert(now.day());
s = sBuff.length()+1;
dmdBuff[s];
sBuff.toCharArray(dmdBuff,s);
dmd.selectFont(System5x7);
dmd.drawString(3,8,dmdBuff,s,0);
```

```
//Display Bulan
sBuff = Convert(now.month());
```



```

s = sBuff.length()+1;
dmdBuff[s];
sBuff.toCharArray(dmdBuff,s);
dmd.selectFont(System5x7);
dmd.drawString(21,8,dmdBuff,s,0);

//Display Tahun
sBuff = Convert(now.year());
s = sBuff.length()+1;
dmdBuff[s];
sBuff.toCharArray(dmdBuff,s);
dmd.selectFont(System5x7);
dmd.drawString(39,8,dmdBuff,s,0);
}

void dept_name()
{
    dmd.clearScreen(true);
    TampilWaktu();
    dmd.selectFont(System5x7);
    dmd.drawString(67,4,"Departemen",10,GRAPHICS_OR);
    delay(1000);
    dmd.clearScreen(true);
    TampilWaktu();
    dmd.selectFont(System5x7);
    dmd.drawString(78,4,"Teknik",6,GRAPHICS_OR);
    delay(1000);
    dmd.clearScreen(true);
    TampilWaktu();
    dmd.selectFont(System5x7);
    dmd.drawString(76,4,"Elektro",7,GRAPHICS_OR);
    delay(1000);
    dmd.clearScreen(true);
    TampilWaktu();
    dmd.selectFont(System5x7);
    dmd.drawString(76,4,"Otomasi",7,GRAPHICS_OR);
    delay(1000);
    for(int b=0;b<5;b++)
    {

```

```

    dmd.clearScreen(true);
    TampilWaktu();
    dmd.selectFont(Arial_14);
    dmd.drawString(75,1,"AA-108",6,GRAPHICS_OR);
    delay(1000);
}
}

```

```

String Convert(int Num){
    if(Num < 10){
        return "0"+String(Num);
    }
    else{
        return String(Num);
    }
}
}

```

## A.7. Datasheets

### \*IC 74HC595

Table 6. Static characteristics

At recommended operating conditions; voltages are referenced to GND (ground = 0 V).

Symbol	Parameter	Conditions	-40 °C to +85 °C			-40 °C to +125 °C		Unit
			Min	Typ	Max	Min	Max	
74HC595								
V <sub>IH</sub>	HIGH-level input voltage	V <sub>CC</sub> = 2.0 V	1.5	1.2	-	1.5	-	V
		V <sub>CC</sub> = 4.5 V	3.15	2.4	-	3.15	-	V
		V <sub>CC</sub> = 6.0 V	4.2	3.2	-	4.2	-	V
V <sub>IL</sub>	LOW-level input voltage	V <sub>CC</sub> = 2.0 V	-	0.8	0.5	-	0.5	V
		V <sub>CC</sub> = 4.5 V	-	2.1	1.35	-	1.35	V
		V <sub>CC</sub> = 6.0 V	-	2.8	1.8	-	1.8	V
V <sub>OH</sub>	HIGH-level output voltage	V <sub>I</sub> = V <sub>IH</sub> or V <sub>IL</sub> all outputs						
		I <sub>O</sub> = -20 µA; V <sub>CC</sub> = 2.0 V	1.9	2.0	-	1.9	-	V
		I <sub>O</sub> = -20 µA; V <sub>CC</sub> = 4.5 V	4.4	4.5	-	4.4	-	V
		I <sub>O</sub> = -20 µA; V <sub>CC</sub> = 6.0 V	5.9	6.0	-	5.9	-	V
		Q7S output						
		I <sub>O</sub> = -4 mA; V <sub>CC</sub> = 4.5 V	3.84	4.32	-	3.7	-	V
		I <sub>O</sub> = -5.2 mA; V <sub>CC</sub> = 6.0 V	5.34	5.81	-	5.2	-	V
		Qn bus driver outputs						
		I <sub>O</sub> = -6 mA; V <sub>CC</sub> = 4.5 V	3.84	4.32	-	3.7	-	V
		I <sub>O</sub> = -7.8 mA; V <sub>CC</sub> = 6.0 V	5.34	5.81	-	5.2	-	V
V <sub>OL</sub>	LOW-level output voltage	V <sub>I</sub> = V <sub>IH</sub> or V <sub>IL</sub> all outputs						
		I <sub>O</sub> = 20 µA; V <sub>CC</sub> = 2.0 V	-	0	0.1	-	0.1	V
		I <sub>O</sub> = 20 µA; V <sub>CC</sub> = 4.5 V	-	0	0.1	-	0.1	V
		I <sub>O</sub> = 20 µA; V <sub>CC</sub> = 6.0 V	-	0	0.1	-	0.1	V
		Q7S output						
		I <sub>O</sub> = 4 mA; V <sub>CC</sub> = 4.5 V	-	0.15	0.33	-	0.4	V
		I <sub>O</sub> = 5.2 mA; V <sub>CC</sub> = 6.0 V	-	0.16	0.33	-	0.4	V
		Qn bus driver outputs						
I <sub>O</sub> = 6 mA; V <sub>CC</sub> = 4.5 V	-	0.15	0.33	-	0.4	V		
I <sub>O</sub> = 7.8 mA; V <sub>CC</sub> = 6.0 V	-	0.16	0.33	-	0.4	V		
I <sub>I</sub>	input leakage current	V <sub>I</sub> = V <sub>CC</sub> or GND; V <sub>CC</sub> = 6.0 V	-	-	±1.0	-	±1.0	µA
I <sub>OZ</sub>	OFF-state output current	V <sub>I</sub> = V <sub>IH</sub> or V <sub>IL</sub> ; V <sub>CC</sub> = 6.0 V; V <sub>O</sub> = V <sub>CC</sub> or GND	-	-	±5.0	-	±10	µA
I <sub>CC</sub>	supply current	V <sub>I</sub> = V <sub>CC</sub> or GND; I <sub>O</sub> = 0 A; V <sub>CC</sub> = 6.0 V	-	-	80	-	160	µA
C <sub>I</sub>	input capacitance		-	3.5	-	-	-	pF

Table 7. Dynamic characteristics

Voltages are referenced to GND (ground = 0 V); for test circuit see [Figure 13](#).

Symbol	Parameter	Conditions	25 °C			-40 °C to +85 °C		-40 °C to +125 °C		Unit
			Min	Typ <sup>[1]</sup>	Max	Min	Max	Min	Max	
74HC595										
t <sub>pd</sub>	propagation delay	SHCP to Q7S; see <a href="#">Figure 8</a> <sup>[2]</sup>								
		V <sub>CC</sub> = 2 V	-	52	160	-	200	-	240	ns
		V <sub>CC</sub> = 4.5 V	-	19	32	-	40	-	48	ns
		V <sub>CC</sub> = 6 V	-	15	27	-	34	-	41	ns
		STCP to Qn; see <a href="#">Figure 9</a> <sup>[2]</sup>								
		V <sub>CC</sub> = 2 V	-	55	175	-	220	-	265	ns
		V <sub>CC</sub> = 4.5 V	-	20	35	-	44	-	53	ns
		V <sub>CC</sub> = 6 V	-	16	30	-	37	-	45	ns
		MR to Q7S; see <a href="#">Figure 11</a> <sup>[2]</sup>								
		V <sub>CC</sub> = 2 V	-	47	175	-	220	-	265	ns
		V <sub>CC</sub> = 4.5 V	-	17	35	-	44	-	53	ns
		V <sub>CC</sub> = 6 V	-	14	30	-	37	-	45	ns
t <sub>en</sub>	enable time	OE to Qn; see <a href="#">Figure 12</a> <sup>[2]</sup>								
		V <sub>CC</sub> = 2 V	-	47	150	-	190	-	225	ns
		V <sub>CC</sub> = 4.5 V	-	17	30	-	38	-	45	ns
		V <sub>CC</sub> = 6 V	-	14	26	-	33	-	38	ns
t <sub>dis</sub>	disable time	OE to Qn; see <a href="#">Figure 12</a> <sup>[2]</sup>								
		V <sub>CC</sub> = 2 V	-	41	150	-	190	-	225	ns
		V <sub>CC</sub> = 4.5 V	-	15	30	-	38	-	45	ns
		V <sub>CC</sub> = 6 V	-	12	27	-	33	-	38	ns
t <sub>w</sub>	pulse width	SHCP HIGH or LOW; see <a href="#">Figure 8</a>								
		V <sub>CC</sub> = 2 V	75	17	-	95	-	110	-	ns
		V <sub>CC</sub> = 4.5 V	15	6	-	19	-	22	-	ns
		V <sub>CC</sub> = 6 V	13	5	-	16	-	19	-	ns
		STCP HIGH or LOW; see <a href="#">Figure 9</a>								
		V <sub>CC</sub> = 2 V	75	11	-	95	-	110	-	ns
		V <sub>CC</sub> = 4.5 V	15	4	-	19	-	22	-	ns
		V <sub>CC</sub> = 6 V	13	3	-	16	-	19	-	ns
		MR LOW; see <a href="#">Figure 11</a>								
		V <sub>CC</sub> = 2 V	75	17	-	95	-	110	-	ns
		V <sub>CC</sub> = 4.5 V	15	6	-	19	-	22	-	ns
		V <sub>CC</sub> = 6 V	13	5	-	16	-	19	-	ns

Table 7. Dynamic characteristics ...continued

Voltages are referenced to GND (ground = 0 V); for test circuit see Figure 13.

Symbol	Parameter	Conditions	25 °C			-40 °C to +85 °C		-40 °C to +125 °C		Unit
			Min	Typ <sup>(1)</sup>	Max	Min	Max	Min	Max	
t <sub>su</sub>	set-up time	DS to SHCP; see Figure 9								
		V <sub>CC</sub> = 2 V	50	11	-	65	-	75	-	ns
		V <sub>CC</sub> = 4.5 V	10	4	-	13	-	15	-	ns
		V <sub>CC</sub> = 6 V	9	3	-	11	-	13	-	ns
		SHCP to STCP; see Figure 10								
		V <sub>CC</sub> = 2 V	75	22	-	95	-	110	-	ns
		V <sub>CC</sub> = 4.5 V	15	8	-	19	-	22	-	ns
		V <sub>CC</sub> = 6 V	13	7	-	16	-	19	-	ns
t <sub>h</sub>	hold time	DS to SHCP; see Figure 10								
		V <sub>CC</sub> = 2 V	3	-6	-	3	-	3	-	ns
		V <sub>CC</sub> = 4.5 V	3	-2	-	3	-	3	-	ns
		V <sub>CC</sub> = 6 V	3	-2	-	3	-	3	-	ns
t <sub>rec</sub>	recovery time	MR to SHCP; see Figure 11								
		V <sub>CC</sub> = 2 V	50	-19	-	65	-	75	-	ns
		V <sub>CC</sub> = 4.5 V	10	-7	-	13	-	15	-	ns
		V <sub>CC</sub> = 6 V	9	-6	-	11	-	13	-	ns
f <sub>max</sub>	maximum frequency	SHCP or STCP; see Figure 8 and 9								
		V <sub>CC</sub> = 2 V	9	30	-	4.8	-	4	-	MHz
		V <sub>CC</sub> = 4.5 V	30	91	-	24	-	20	-	MHz
		V <sub>CC</sub> = 6 V	35	108	-	28	-	24	-	MHz
C <sub>PD</sub>	power dissipation capacitance	f <sub>i</sub> = 1 MHz; V <sub>i</sub> = GND to V <sub>CC</sub> <sup>(5)(7)</sup>	-	115	-	-	-	-	-	pF

# **\*IC 74HC138**

**Table 6. Static characteristics**

At recommended operating conditions; voltages are referenced to GND (ground = 0 V).

Symbol	Parameter	Conditions	T <sub>amb</sub> = 25 °C			T <sub>amb</sub> = -40 °C to +85 °C		T <sub>amb</sub> = -40 °C to +125 °C		Unit
			Min	Typ	Max	Min	Max	Min	Max	
74HC138										
V <sub>IH</sub>	HIGH-level input voltage	V <sub>CC</sub> = 2.0 V	1.5	1.2	-	1.5	-	1.5	-	V
		V <sub>CC</sub> = 4.5 V	3.15	2.4	-	3.15	-	3.15	-	V
		V <sub>CC</sub> = 6.0 V	4.2	3.2	-	4.2	-	4.2	-	V
V <sub>IL</sub>	LOW-level input voltage	V <sub>CC</sub> = 2.0 V	-	0.8	0.5	-	0.5	-	0.5	V
		V <sub>CC</sub> = 4.5 V	-	2.1	1.35	-	1.35	-	1.35	V
		V <sub>CC</sub> = 6.0 V	-	2.8	1.8	-	1.8	-	1.8	V
V <sub>OH</sub>	HIGH-level output voltage	V <sub>I</sub> = V <sub>IH</sub> or V <sub>IL</sub>								
		I <sub>O</sub> = -20 μA; V <sub>CC</sub> = 2.0 V	1.9	2.0	-	1.9	-	1.9	-	V
		I <sub>O</sub> = -20 μA; V <sub>CC</sub> = 4.5 V	4.4	4.5	-	4.4	-	4.4	-	V
		I <sub>O</sub> = -20 μA; V <sub>CC</sub> = 6.0 V	5.9	6.0	-	5.9	-	5.9	-	V
		I <sub>O</sub> = -4.0 mA; V <sub>CC</sub> = 4.5 V	3.98	4.32	-	3.84	-	3.7	-	V
		I <sub>O</sub> = -5.2 mA; V <sub>CC</sub> = 6.0 V	5.48	5.81	-	5.34	-	5.2	-	V
V <sub>OL</sub>	LOW-level output voltage	V <sub>I</sub> = V <sub>IH</sub> or V <sub>IL</sub>								
		I <sub>O</sub> = 20 μA; V <sub>CC</sub> = 2.0 V	-	0	0.1	-	0.1	-	0.1	V
		I <sub>O</sub> = 20 μA; V <sub>CC</sub> = 4.5 V	-	0	0.1	-	0.1	-	0.1	V
		I <sub>O</sub> = 20 μA; V <sub>CC</sub> = 6.0 V	-	0	0.1	-	0.1	-	0.1	V
		I <sub>O</sub> = 4.0 mA; V <sub>CC</sub> = 4.5 V	-	0.15	0.26	-	0.33	-	0.4	V
		I <sub>O</sub> = 5.2 mA; V <sub>CC</sub> = 6.0 V	-	0.16	0.26	-	0.33	-	0.4	V
I <sub>I</sub>	input leakage current	V <sub>I</sub> = V <sub>CC</sub> or GND; V <sub>CC</sub> = 6.0 V	-	-	±0.1	-	±1.0	-	±1.0	μA
I <sub>CC</sub>	supply current	V <sub>I</sub> = V <sub>CC</sub> or GND; I <sub>O</sub> = 0 A; V <sub>CC</sub> = 6.0 V	-	-	8.0	-	80	-	160	μA

**Table 6. Static characteristics ...continued**

At recommended operating conditions; voltages are referenced to GND (ground = 0 V).

Symbol	Parameter	Conditions	T <sub>amb</sub> = 25 °C			T <sub>amb</sub> = -40 °C to +85 °C		T <sub>amb</sub> = -40 °C to +125 °C		Unit
			Min	Typ	Max	Min	Max	Min	Max	
C <sub>I</sub>	input capacitance		-	3.5	-					pF

Table 7. Dynamic characteristics

Voltages are referenced to GND (ground = 0 V);  $C_L = 50$  pF unless otherwise specified; for test circuit see Figure 8.

Symbol	Parameter	Conditions	T <sub>amb</sub> = 25 °C			T <sub>amb</sub> = -40 °C to +85 °C		T <sub>amb</sub> = -40 °C to +125 °C		Unit
			Min	Typ	Max	Min	Max	Min	Max	
74HC138										
t <sub>pd</sub>	propagation delay	An to $\overline{Y}_n$ ; see <a href="#">Figure 6</a> <a href="#">[1]</a>								
		V <sub>CC</sub> = 2.0 V	-	41	150	-	190	-	225	ns
		V <sub>CC</sub> = 4.5 V	-	15	30	-	38	-	45	ns
		V <sub>CC</sub> = 5 V; C <sub>L</sub> = 15 pF	-	12	-	-	-	-	-	ns
		V <sub>CC</sub> = 6.0 V	-	12	26	-	33	-	38	ns
		E3 to $\overline{Y}_n$ ; see <a href="#">Figure 6</a> <a href="#">[1]</a>								
		V <sub>CC</sub> = 2.0 V	-	47	150	-	190	-	225	ns
		V <sub>CC</sub> = 4.5 V	-	17	20	-	38	-	45	ns
		V <sub>CC</sub> = 5 V; C <sub>L</sub> = 15 pF	-	14	-	-	-	-	-	ns
		V <sub>CC</sub> = 6.0 V	-	14	26	-	33	-	38	ns
		$\overline{E}_n$ to $\overline{Y}_n$ ; see <a href="#">Figure 7</a> <a href="#">[1]</a>								
		V <sub>CC</sub> = 2.0 V	-	47	150	-	190	-	225	ns
		V <sub>CC</sub> = 4.5 V	-	17	20	-	38	-	45	ns
		V <sub>CC</sub> = 5 V; C <sub>L</sub> = 15 pF	-	14	-	-	-	-	-	ns
		V <sub>CC</sub> = 6.0 V	-	14	26	-	33	-	38	ns
t <sub>t</sub>	transition time	$\overline{Y}_n$ ; see <a href="#">Figure 6</a> and <a href="#">Figure 7</a> <a href="#">[2]</a>								
		V <sub>CC</sub> = 2.0 V	-	19	75	-	95	-	110	ns
		V <sub>CC</sub> = 4.5 V	-	7	15	-	19	-	22	ns
		V <sub>CC</sub> = 6.0 V	-	6	13	-	16	-	19	ns
C <sub>PD</sub>	power dissipation capacitance	C <sub>L</sub> = 50 pF; f = 1 MHz; V <sub>I</sub> = GND to V <sub>CC</sub> <a href="#">[3]</a>	-	67	-	-	-	-	pF	

## \*IC 74HC245

Table 7: Static characteristics type 74HC245

At recommended operating conditions; voltages are referenced to GND (ground = 0 V).

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
T <sub>amb</sub> = 25 °C						
V <sub>IH</sub>	HIGH-level input voltage	V <sub>CC</sub> = 2.0 V	1.5	1.2	-	V
		V <sub>CC</sub> = 4.5 V	3.15	2.4	-	V
		V <sub>CC</sub> = 6.0 V	4.2	3.2	-	V
V <sub>IL</sub>	LOW-level input voltage	V <sub>CC</sub> = 2.0 V	-	0.8	0.5	V
		V <sub>CC</sub> = 4.5 V	-	2.1	1.35	V
		V <sub>CC</sub> = 6.0 V	-	2.8	1.8	V
V <sub>OH</sub>	HIGH-level output voltage	V <sub>I</sub> = V <sub>IH</sub> or V <sub>IL</sub>				
		I <sub>O</sub> = -20 μA; V <sub>CC</sub> = 2.0 V	1.9	2.0	-	V
		I <sub>O</sub> = -20 μA; V <sub>CC</sub> = 4.5 V	4.4	4.5	-	V
		I <sub>O</sub> = -20 μA; V <sub>CC</sub> = 6.0 V	5.9	6.0	-	V
		I <sub>O</sub> = -6.0 mA; V <sub>CC</sub> = 4.5 V	3.98	4.32	-	V
		I <sub>O</sub> = -7.8 mA; V <sub>CC</sub> = 6.0 V	5.48	5.81	-	V

Table 7: Static characteristics type 74HC245 ...continued

At recommended operating conditions; voltages are referenced to GND (ground = 0 V).

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{OL}$	LOW-level output voltage	$V_I = V_{IH}$ or $V_{IL}$				
		$I_O = 20\ \mu\text{A}; V_{CC} = 2.0\ \text{V}$	-	0	0.1	V
		$I_O = 20\ \mu\text{A}; V_{CC} = 4.5\ \text{V}$	-	0	0.1	V
		$I_O = 20\ \mu\text{A}; V_{CC} = 6.0\ \text{V}$	-	0	0.1	V
		$I_O = 6.0\ \text{mA}; V_{CC} = 4.5\ \text{V}$	-	0.15	0.26	V
		$I_O = 7.8\ \text{mA}; V_{CC} = 6.0\ \text{V}$	-	0.16	0.26	V
$I_{LI}$	input leakage current	$V_I = V_{CC}$ or GND; $V_{CC} = 6.0\ \text{V}$	-	-	$\pm 0.1$	$\mu\text{A}$
$I_{OZ}$	OFF-state output current	$V_I = V_{IH}$ or $V_{IL}; V_O = V_{CC}$ or GND; $V_{CC} = 6.0\ \text{V}$	-	-	$\pm 0.5$	$\mu\text{A}$
$I_{CC}$	quiescent supply current	$V_I = V_{CC}$ or GND; $I_O = 0\ \text{A};$ $V_{CC} = 6.0\ \text{V}$	-	-	8.0	$\mu\text{A}$
$C_I$	input capacitance		-	3.5	-	pF
$C_{IO}$	input/output capacitance		-	10	-	pF
$T_{amb} = -40\ ^\circ\text{C}$ to $+85\ ^\circ\text{C}$						
$V_{IH}$	HIGH-level input voltage	$V_{CC} = 2.0\ \text{V}$	1.5	-	-	V
		$V_{CC} = 4.5\ \text{V}$	3.15	-	-	V
		$V_{CC} = 6.0\ \text{V}$	4.2	-	-	V
$V_{IL}$	LOW-level input voltage	$V_{CC} = 2.0\ \text{V}$	-	-	0.5	V
		$V_{CC} = 4.5\ \text{V}$	-	-	1.35	V
		$V_{CC} = 6.0\ \text{V}$	-	-	1.8	V
$V_{OH}$	HIGH-level output voltage	$V_I = V_{IH}$ or $V_{IL}$				
		$I_O = -20\ \mu\text{A}; V_{CC} = 2.0\ \text{V}$	1.9	-	-	V
		$I_O = -20\ \mu\text{A}; V_{CC} = 4.5\ \text{V}$	4.4	-	-	V
		$I_O = -20\ \mu\text{A}; V_{CC} = 6.0\ \text{V}$	5.9	-	-	V
		$I_O = -6.0\ \text{mA}; V_{CC} = 4.5\ \text{V}$	3.84	-	-	V
		$I_O = -7.8\ \text{mA}; V_{CC} = 6.0\ \text{V}$	5.34	-	-	V
$V_{OL}$	LOW-level output voltage	$V_I = V_{IH}$ or $V_{IL}$				
		$I_O = 20\ \mu\text{A}; V_{CC} = 2.0\ \text{V}$	-	-	0.1	V
		$I_O = 20\ \mu\text{A}; V_{CC} = 4.5\ \text{V}$	-	-	0.1	V
		$I_O = 20\ \mu\text{A}; V_{CC} = 6.0\ \text{V}$	-	-	0.1	V
		$I_O = 6.0\ \text{mA}; V_{CC} = 4.5\ \text{V}$	-	-	0.33	V
		$I_O = 7.8\ \text{mA}; V_{CC} = 6.0\ \text{V}$	-	-	0.33	V
$I_{LI}$	input leakage current	$V_I = V_{CC}$ or GND; $V_{CC} = 6.0\ \text{V}$	-	-	$\pm 1.0$	$\mu\text{A}$
$I_{OZ}$	OFF-state output current	$V_I = V_{IH}$ or $V_{IL}; V_O = V_{CC}$ or GND; $V_{CC} = 6.0\ \text{V}$	-	-	$\pm 5.0$	$\mu\text{A}$
$I_{CC}$	quiescent supply current	$V_I = V_{CC}$ or GND; $I_O = 0\ \text{A};$ $V_{CC} = 6.0\ \text{V}$	-	-	80	$\mu\text{A}$
$T_{amb} = -40\ ^\circ\text{C}$ to $+125\ ^\circ\text{C}$						
$V_{IH}$	HIGH-level input voltage	$V_{CC} = 2.0\ \text{V}$	1.5	-	-	V
		$V_{CC} = 4.5\ \text{V}$	3.15	-	-	V
		$V_{CC} = 6.0\ \text{V}$	4.2	-	-	V



Table 7: Static characteristics type 74HC245 ...continued

At recommended operating conditions; voltages are referenced to GND (ground = 0 V).

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{IL}$	LOW-level input voltage	$V_{CC} = 2.0\text{ V}$	-	-	0.5	V
		$V_{CC} = 4.5\text{ V}$	-	-	1.35	V
		$V_{CC} = 6.0\text{ V}$	-	-	1.8	V
$V_{OH}$	HIGH-level output voltage	$V_I = V_{IH}$ or $V_{IL}$	-	-	-	
		$I_O = -20\text{ }\mu\text{A}; V_{CC} = 2.0\text{ V}$	1.9	-	-	V
		$I_O = -20\text{ }\mu\text{A}; V_{CC} = 4.5\text{ V}$	4.4	-	-	V
		$I_O = -20\text{ }\mu\text{A}; V_{CC} = 6.0\text{ V}$	5.9	-	-	V
		$I_O = -8.0\text{ mA}; V_{CC} = 4.5\text{ V}$	3.7	-	-	V
		$I_O = -7.8\text{ mA}; V_{CC} = 6.0\text{ V}$	5.2	-	-	V
$V_{OL}$	LOW-level output voltage	$V_I = V_{IH}$ or $V_{IL}$	-	-	-	
		$I_O = 20\text{ }\mu\text{A}; V_{CC} = 2.0\text{ V}$	-	-	0.1	V
		$I_O = 20\text{ }\mu\text{A}; V_{CC} = 4.5\text{ V}$	-	-	0.1	V
		$I_O = 20\text{ }\mu\text{A}; V_{CC} = 6.0\text{ V}$	-	-	0.1	V
		$I_O = 6.0\text{ mA}; V_{CC} = 4.5\text{ V}$	-	-	0.4	V
		$I_O = 7.8\text{ mA}; V_{CC} = 6.0\text{ V}$	-	-	0.4	V
$I_{II}$	input leakage current	$V_I = V_{CC}$ or GND; $V_{CC} = 6.0\text{ V}$	-	-	$\pm 1.0$	$\mu\text{A}$
$I_{OZ}$	OFF-state output current	$V_I = V_{IH}$ or $V_{IL}; V_O = V_{CC}$ or GND; $V_{CC} = 6.0\text{ V}$	-	-	$\pm 10.0$	$\mu\text{A}$
$I_{CC}$	quiescent supply current	$V_I = V_{CC}$ or GND; $I_O = 0\text{ A};$ $V_{CC} = 6.0\text{ V}$	-	-	160	$\mu\text{A}$

Table 9: Dynamic characteristics type 74HC245

GND = 0 V; test circuit see Figure 7.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$T_{amb} = 25\text{ }^\circ\text{C}$						
$t_{PHL}, t_{PLH}$	propagation delay An to Bn or Bn to An	see Figure 5				
		$V_{CC} = 2.0\text{ V}$	-	25	90	ns
		$V_{CC} = 4.5\text{ V}$	-	9	18	ns
		$V_{CC} = 5.0\text{ V}; C_L = 15\text{ pF}$	-	7	-	ns
		$V_{CC} = 6.0\text{ V}$	-	7	15	ns
$t_{PZH}, t_{PZL}$	3-state output enable time $\overline{OE}$ to An or OE to Bn	see Figure 6				
		$V_{CC} = 2.0\text{ V}$	-	30	150	ns
		$V_{CC} = 4.5\text{ V}$	-	11	30	ns
		$V_{CC} = 6.0\text{ V}$	-	9	26	ns
$t_{PHZ}, t_{PLZ}$	3-state output disable time $\overline{OE}$ to An or OE to Bn	see Figure 6				
		$V_{CC} = 2.0\text{ V}$	-	41	150	ns
		$V_{CC} = 4.5\text{ V}$	-	15	30	ns
		$V_{CC} = 6.0\text{ V}$	-	12	26	ns
$t_{THL}, t_{TLH}$	output transition time	see Figure 5				
		$V_{CC} = 2.0\text{ V}$	-	14	60	ns
		$V_{CC} = 4.5\text{ V}$	-	5	12	ns
		$V_{CC} = 6.0\text{ V}$	-	4	10	ns
$C_{PD}$	power dissipation capacitance per transceiver	$V_I = \text{GND to } V_{CC}$	①	30	-	pF
$T_{amb} = -40\text{ }^\circ\text{C to } +85\text{ }^\circ\text{C}$						
$t_{PHL}, t_{PLH}$	propagation delay An to Bn or Bn to An	see Figure 5				
		$V_{CC} = 2.0\text{ V}$	-	-	115	ns
		$V_{CC} = 4.5\text{ V}$	-	-	23	ns
		$V_{CC} = 6.0\text{ V}$	-	-	20	ns

Table 9: Dynamic characteristics type 74HC245 ...continued  
 GND = 0 V; test circuit see Figure 7.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$t_{PZH}, t_{PZL}$	3-state output enable time $\overline{OE}$ to An or $\overline{OE}$ to Bn	see Figure 8				
		$V_{CC} = 2.0 \text{ V}$	-	-	190	ns
		$V_{CC} = 4.5 \text{ V}$	-	-	38	ns
		$V_{CC} = 6.0 \text{ V}$	-	-	33	ns
$t_{PHZ}, t_{PLZ}$	3-state output disable time $\overline{OE}$ to An or $\overline{OE}$ to Bn	see Figure 8				
		$V_{CC} = 2.0 \text{ V}$	-	-	190	ns
		$V_{CC} = 4.5 \text{ V}$	-	-	38	ns
		$V_{CC} = 6.0 \text{ V}$	-	-	33	ns
$t_{rHL}, t_{rLH}$	output transition time	see Figure 5				
		$V_{CC} = 2.0 \text{ V}$	-	-	75	ns
		$V_{CC} = 4.5 \text{ V}$	-	-	15	ns
		$V_{CC} = 6.0 \text{ V}$	-	-	13	ns
$T_{amb} = -40\text{ }^{\circ}\text{C to } +125\text{ }^{\circ}\text{C}$						
$t_{PHL}, t_{PLH}$	propagation delay An to Bn or Bn to An	see Figure 5				
		$V_{CC} = 2.0 \text{ V}$	-	-	135	ns
		$V_{CC} = 4.5 \text{ V}$	-	-	27	ns
		$V_{CC} = 6.0 \text{ V}$	-	-	23	ns
$t_{PZH}, t_{PZL}$	3-state output enable time $\overline{OE}$ to An or $\overline{OE}$ to Bn	see Figure 8				
		$V_{CC} = 2.0 \text{ V}$	-	-	225	ns
		$V_{CC} = 4.5 \text{ V}$	-	-	45	ns
		$V_{CC} = 6.0 \text{ V}$	-	-	38	ns
$t_{PHZ}, t_{PLZ}$	3-state output disable time $\overline{OE}$ to An or $\overline{OE}$ to Bn	see Figure 8				
		$V_{CC} = 2.0 \text{ V}$	-	-	225	ns
		$V_{CC} = 4.5 \text{ V}$	-	-	45	ns
		$V_{CC} = 6.0 \text{ V}$	-	-	38	ns
$t_{rHL}, t_{rLH}$	output transition time	see Figure 5				
		$V_{CC} = 2.0 \text{ V}$	-	-	90	ns
		$V_{CC} = 4.5 \text{ V}$	-	-	18	ns
		$V_{CC} = 6.0 \text{ V}$	-	-	15	ns

[1]  $C_{PD}$  is used to determine the dynamic power dissipation ( $P_D$  in  $\mu\text{W}$ ):

$$P_D = C_{PD} \times V_{CC}^2 \times f_i \times N + \sum (C_L \times V_{CC}^2 \times f_o) \text{ where:}$$

$f_i$  = input frequency in MHz;

$f_o$  = output frequency in MHz;

$C_L$  = output load capacitance in pF;

$V_{CC}$  = supply voltage in V;

$N$  = number of inputs switching;

$\sum (C_L \times V_{CC}^2 \times f_o)$  = sum of outputs.

# \*IC APM4953

## Electrical Characteristics (T<sub>A</sub>=25°C unless otherwise noted)

Symbol	Parameter	Test Condition	APM4953			Unit
			Min.	Typ <sup>a</sup>	Max.	
Static						
BV <sub>DSS</sub>	Drain-Source Breakdown Voltage	V <sub>GS</sub> =0V, I <sub>DS</sub> =-250μA	-30			V
I <sub>DSS</sub>	Zero Gate Voltage Drain Current	V <sub>DS</sub> =-24V, V <sub>GS</sub> =0V			-1	μA
V <sub>GS(th)</sub>	Gate Threshold Voltage	V <sub>DS</sub> =V <sub>GS</sub> , I <sub>DS</sub> =-250μA	-1	-1.5	-2	V
I <sub>GSS</sub>	Gate Leakage Current	V <sub>GS</sub> =±25V, V <sub>DS</sub> =0V			±100	nA
R <sub>DS(ON)</sub>	Drain-Source On-state Resistance <sup>b</sup>	V <sub>GS</sub> =-10V, I <sub>DS</sub> =-4.9A		53	60	mΩ
		V <sub>GS</sub> =-4.5V, I <sub>DS</sub> =-3.6A		80	95	
V <sub>SD</sub>	Diode Forward Voltage <sup>b</sup>	I <sub>SD</sub> =-1.7A, V <sub>GS</sub> =0V		-0.7	-1.3	V
Dynamic <sup>a</sup>						
Q <sub>g</sub>	Total Gate Charge	V <sub>DS</sub> =-15V, I <sub>GS</sub> =-10V		22.3	29	nC
Q <sub>gs</sub>	Gate-Source Charge	I <sub>g</sub> =-4.6A		4.65		
Q <sub>gd</sub>	Gate-Drain Charge			2		
t <sub>d(ON)</sub>	Turn-on Delay Time	V <sub>DD</sub> =-15V, I <sub>g</sub> =-2A,		10	18	ns
T <sub>r</sub>	Turn-on Rise Time	V <sub>GEN</sub> =-10V, R <sub>G</sub> =6Ω		15	20	
t <sub>d(OFF)</sub>	Turn-off Delay Time	R <sub>L</sub> =7.5Ω		22	38	
T <sub>f</sub>	Turn-off Fall Time			15	25	
C <sub>iss</sub>	Input Capacitance	V <sub>GS</sub> =0V		1260		pF
C <sub>OSS</sub>	Output Capacitance	V <sub>DS</sub> =-25V		340		
C <sub>rss</sub>	Reverse Transfer Capacitance	Frequency=1.0MHz		220		

### Notes

<sup>a</sup>: Pulse test; pulse width ≤300μs, duty cycle ≤ 2%

<sup>b</sup>: Guaranteed by design, not subject to production testing

# \*IC SN74HC04

## 6.4 Electrical Characteristics

over operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS	V <sub>CC</sub>	T <sub>A</sub> = 25°C			SN54HC04		SN74HC04		UNIT
			MIN	TYP	MAX	MIN	MAX	MIN	MAX	
V <sub>OH</sub>	V <sub>I</sub> = V <sub>IH</sub> or V <sub>IL</sub>	I <sub>OH</sub> = -20 μA	2 V	1.9	1.998	1.9		1.9		V
			4.5 V	4.4	4.499	4.4		4.4		
			6 V	5.9	5.999	5.9		5.9		
		I <sub>OH</sub> = -4 mA	4.5 V	3.98	4.3	3.7		3.84		
		I <sub>OH</sub> = -5.2 mA	6 V	5.48	5.8	5.2		5.34		
V <sub>OL</sub>	V <sub>I</sub> = V <sub>IH</sub> or V <sub>IL</sub>	I <sub>OL</sub> = 20 μA	2 V		0.002 0.1		0.1		0.1	V
			4.5 V		0.001 0.1		0.1		0.1	
			6 V		0.001 0.1		0.1		0.1	
		I <sub>OL</sub> = 4 mA	4.5 V		0.17 0.26		0.4		0.33	
		I <sub>OL</sub> = 5.2 mA	6 V		0.15 0.26		0.4		0.33	
I <sub>I</sub>	V <sub>I</sub> = V <sub>CC</sub> or 0	6 V		±0.1	±100		±1000		±1000	nA
I <sub>CC</sub>	V <sub>I</sub> = V <sub>CC</sub> or 0, I <sub>O</sub> = 0	6 V			2		40		20	μA
C <sub>i</sub>		6 V			3 10		10		10	pF

## \*XBee S2C

### Performance specifications

This table describes the performance specifications for the devices.

Specification	XBee ZigBee S2C	XBee-PRO ZigBee S2C	XBee ZigBee S2D
Indoor/urban range	Up to 60 m (200 ft)	Up to 90 m (300 ft)	Up to 60 m (200 ft)
Outdoor RF line-of-sight range	Up to 1200 m (4000 ft)	Up to 3200 m (2 mi)	Up to 1200 m (4000 ft)
Transmit power output (maximum)	6.3 mW (+8 dBm), boost mode 3.1 mW (+5 dBm), normal mode channel 26 max power is +3 dBm	63 mW (+18 dBm)	6.3 mW (+8 dBm) channel 26 max power is +1 dBm
RF data rate	250,000 b/s		
Receiver sensitivity	-102 dBm, boost mode -100 dBm, normal mode	-101 dBm	-102 dBm, boost mode -100 dBm, normal mode

### Power requirements

The following table describes the power requirements for the devices.

Specification	XBee ZigBee S2C	XBee-PRO ZigBee S2C	XBee ZigBee S2D
Adjustable power	Yes		
Supply voltage	2.1 - 3.6 V 2.2 - 3.6 V for programmable version	2.7 - 3.6 V	2.1 - 3.6 V
Operating current (transmit)	45 mA (+8 dBm, boost mode) 33 mA (+5 dBm, normal mode)	120 mA @ +3.3 V, +18 dBm	45 mA
Operating current (receive)	31 mA (boost mode) 28 mA (normal mode)	31 mA	31 mA
Power-down current	< 1 $\mu$ A @ 25°C		< 3 $\mu$ A @ 25°C

### General specifications

The following table describes the general specifications for the devices.

Specification	XBee ZigBee S2C	XBee-PRO ZigBee S2C	XBee ZigBee S2D
Operating frequency band	ISM 2.4 - 2.5 GHz		
Form factor	through-hole, surface-mount		surface-mount

Specification	XBee ZigBee S2C	XBee-PRO ZigBee S2C	XBee ZigBee S2D
Dimensions	through-hole: 2.438 x 2.761 cm (0.960 x 1.087 in) surface-mount: 2.199 x 3.4 x 0.305 cm (0.866 x 1.33 x 0.120 in)	through-hole: 2.438 x 3.294 cm (0.960 x 1.297 in) surface-mount: 2.199 x 3.4 x 0.305 cm (0.866 x 1.33 x 0.120 in)	surface-mount: 2.199 x 3.4 x 0.305 cm (0.866 x 1.33 x 0.120 in)
Operating temperature	-40 to 85 °C (industrial)		
Antenna options	through-hole: PCB antenna, U.FL connector, RPSMA connector, or integrated wire surface-mount: RF pad, PCB antenna, or U.FL connector		

## Networking and security specifications

The following table describes the networking and security specifications for the devices.

Specification	XBee ZigBee S2C	XBee-PRO ZigBee S2C	XBee ZigBee S2D
Supported network topologies	Point-to-point, point-to-multipoint, peer-to-peer, and DigiMesh		
Number of channels	16 Direct sequence channels	15 Direct sequence channels	16 Direct sequence channels
Interface immunity	Direct Sequence Spread Spectrum (DSSS)		
Channels	11 to 26		
Addressing options	PAN ID and addresses, cluster IDs and endpoints (optional)		

\

-----Halaman ini sengaja dikosongkan-----

## RIWAYAT HIDUP PENULIS



Nama : Awataraning Tunjung P.  
TTL : Magetan, 10 April 1996  
Jenis Kelamin : Laki-laki  
Agama : Islam  
Alamat Rumah : Jl. H. Moestalin A-19  
P. Bun, Kotawaringin  
Barat  
Telp/HP : 085787578676  
E-mail :  
awataraningtunjungpradip  
ta@gmail.com  
Hobi : Olahraga, *travelling*,  
menulis

### RIWAYAT PENDIDIKAN

- 2002-2008 : SDN 1 Sidorejo
- 2008-2011 : SMP Negeri 1 Arut Selatan
- 2011-2014 : SMA Negeri 1 Pangkalan Bun
- 2014 – sekarang : Program Studi Komputer Kontrol,  
Departemen Teknik Elektro Otomasi, ITS

### PENGALAMAN KERJA

- Kerja Praktek di PT. CNC CONTROLLER INDONESIA Bekasi (2016)

### PENGALAMAN ORGANISASI

- Staff Dewan Perwakilan Mahasiswa(DPM) ITS (2015)
- Staff Departemen Sosial Masyarakat (SOSMAS) HIMAD3TEKTRO ITS (2015-2016)
- Kepala Badan Koordinasi Pemandu Departemen Pengembangan Sumber Daya Mahasiswa (PSDM) HIMA D3 Teknik Elektro ITS (2016 – 2017)
- Asisten Laboratorium Elektronika Dasar D3 Teknik Elektro ITS (2016 – 2017)

#### PENGALAMAN KEPANITIAAN

- Latihan Keterampilan Manajemen Mahasiswa Pra-Tingkat Dasar BEM FTI- ITS (2015 dan 2016)
- Latihan Keterampilan Manajemen Mahasiswa Tingkat Dasar HIMAD3TEKTRO ITS (2015, 2016, 2017)
- Pelatihan Pemandu Latihan Keterampilan Manajemen Mahasiswa BEM FTI-ITS (2016)
- Industrial Automation and Robotics Competition (2015 dan 2016)
- FTI Olympic Games (FOG) BEM FTI-ITS (2015)